Dispatchability Conditions for Schedules with Consumable Resources

Richard J. Wallace and **Eugen C. Freuder** [1]

Abstract. Earlier work on scheduling by autonomous systems has demonstrated that schedules in the form of simple temporal networks, with intervals of values for possible event-times, can be made "dispatchable", i.e., executable incrementally in real time with guarantees against failure due to unfortunate event-time selections. In the present work we show how the property of dispatchability can be extended to networks that include constraints for consumable resources. We describe conditions that can be placed on resource use by activities during execution to avoid oversubscription while insuring schedule dispatchability. This involves preventing interactions between resource use and scheduling per se, that can compromise dispatchability. We also describe strategies that can be used in conjunction with these conditions to increase flexibility of resource allocation. This work indicates that flexible handling of resource use can be safely extended to the execution layer to provide more effective deployment of consumable resources.

1 Introduction

Like many other autonomous systems, the scheduling system within of the Remote Agent architecture developed at NASA-Ames [5] has two basic layers of operation. The first is the Planner-Scheduler which produces a basic schedule of operation [3]. The output from this subsystem consists of an envelope of scheduling times for each operation, or activity, which allows some flexibility in scheduling specific activities based on current conditions. The task of selecting specific times for operation is carried out in a second layer by the plan executor, or "Executive", which issues the command signals for operating the physical system and monitors the progress of the system to determine whether activites have been performed successfully.

Because the Executive is operating in real time, the constraints on its operation are severe. In particular, during the instantiation of a schedule the Executive cannot afford to backtrack. That is, it cannot go back and reschedule earlier activities whenever its previous decisions have caused it to reach a point where there are no options (a 'dead end'), because these earlier activities have already begun. For this reason, when actual plan execution begins there must be guarantees that a schedule derived from the time-envelopes is executable incrementally or "dispatchable". In recent work it has been shown that consistent temporal constraint networks, which are a basic component of the Planner-Scheduler's output, can be made dispatchable [4].

In this paper we describe work we have done to extend the earlier work on dispatchable networks of temporal constraints to situations where there are also *consumable* resources. The goal of this work is to establish conditions on resource use that allow us to guarantee that a resource will not be oversubscribed, while still maintaining guarantees for schedule dispatchability. A critical problem in this case is interaction between decisions regarding resource use and selection of start- and end-times of activities in the schedule. To solve this, we derive a basic condition on resource use to which are added qualifications that serve to avoid unwonted interaction. This paper offers an alternative and in some ways more general approach to one presented in an earlier paper [6].

The next section discusses dispatchable execution and the incorporation of consumable resources into this process. Section 3 describes conditions for avoiding oversubscription of a consumable resource and means of avoiding interactions between restrictions on resource use and maintaining schedule dispatchability. Building on the previous section, Section 4 describes a general condition for dispatchability. Section 5 discusses strategies that can be used for increasing flexibility of resource use while still meeting the condition for dispatchability. Section 6 gives conclusions.

2 Incorporating Resource Use at the Executive Level

In the present scheduling system, the Planner-Scheduler sends the Executive an envelope of acceptable scheduling times in the form of a simple temporal network (STN) [1]. A critical characteristic of this network is that it is not only consistent but *dispatchable*. This means that events can be instantiated successively, whenever the current time is within the present bounds of an event, and the system will not encounter a dead-end, i.e., an event that cannot be scheduled because the current time is beyond its present time bounds [4].

The basic dispatching execution proceeds as follows (see [4] for details). At each step of execution an event x is selected from a pool of candidate events whose antecedents have already been instantiated, and whose time bounds bracket the current time. After this event has been set to the current time, constraint propagation occurs, which is restricted to adjacent nodes in the network. In the following example, based on the STN in Figure 1, instantiations are shown on the left and results of propagation on the right, in terms of acceptable intervals for events whose nodes are adjacent in the constraint graph.

a = 0	b = $4-9$, c = $4-6$
c = 5	e = $9-12$
b = 7	d = $9-11$
d = 9	e = $9-12$, f = $16-19$, g = $16-19$
e = 12	f = $17-19$, g = $17-19$
g = 17	f = 17
f = 17	

[1] Constraint Computation Center, University of New Hampshire, Durham, NH 03824

© 2000 R.J. Wallace and E.C. Freuder
ECAI 2000. *14th European Conference on Artificial Intelligence*
Edited by Werner Horn
Published in 2000 by IOS Press

In the situation we wish to consider, some of the activities in the schedule are associated with use of a consumable resource. As an example, consider the storage of information from sensing devices on a spacecraft, which normally involves an onboard solid state recorder (SSR). Data is stored in the SSR until the craft is in communication with the Deep Space Network on earth, when the information is downloaded. Data storage occurs during a Record state; in this state resource capacity is used up. Data transmission occurs during a Playback state, when information is relayed to earth; in this state resource capacity is released. The SSR is, therefore, a consumable resource with a fixed capacity that cannot be exceeded during the schedule. However, in this case the resource can be renewed periodically. We refer to the set of resource activities between renewals (here, Playbacks) as a "bout" of resource use. In the present paper we will focus on single bouts. Sequences of bouts require adjustments of the capacity value, and are treated in detail in [6]. Note that in many cases, such as power use, there may only be one bout.

The manner of representing resource use at the Executive level will depend on the way in which we establish conditions for dispatchability, as will be seen below. For expository purposes we will represent resource use as a separate constraint network, based on constraints of the form $x_1 + x_2 ... \leq$ C, where x_i is an instance of resource use and C is the resource capacity, which we call the consumable resource network, or cRN. An example is shown in the upper part of Figure 1, which shows a cRN for a single consumable resource and a single bout of use. In this graph, intervals represent bounds on resource use for a given activity. Thus, each interval, [10,20], represents a range of possible use of a resource between 10 and 20 units. In addition, a single k-ary constraint between endpoints prevents the resource capacity (here, 30 units) from being exceeded during the bout.

In the figure, cRN nodes are linked to STN nodes that correspond to the same activity. The linkage between the STN and cRN is indicated by dashed lines, each labeled with its constant of proportionality. These links are not constraints, but instead link two representations of the same activity - as a temporal duration and as an instance of resource use. Here (and in general), resource use is assumed to be a nondecreasing function of time. For expository purposes we will assume a linear relation, specifically, multiplication of the temporal bounds by a positive or negative quantity for resource use and release, respectively, this quantity being constant for a given activity. Of importance is the fact that the mapping from activity duration to associated resource use is both bijective, i.e., one-to-one and onto, and monotonic.

3 The Dispatchability Problem for the Executive STN-cRN

We now consider what it means to extend the notion of dispatchability to schedules with resource constraints. This extension entails two requirements:

1. resource capacity cannot be exceeded, even momentarily
2. the dispatchability of the schedule must be maintained

In addition, if possible we would like to extend the notion of flexibility of execution to resource use as well as event times.

At the outset, it is important to note that the tractability of the elaborated problem, represented here as an STN-cRN is not a basic issue. This is because both temporal and consumable resource constraints (represented as less-than inequalities) are closed under a binary function, min, that is associative, commutative, and idempotent. This CSP, therefore, falls under tractability class 2 of [2].

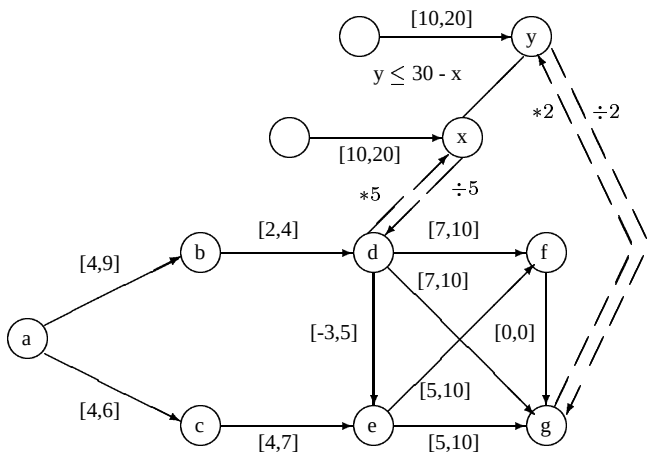

Figure 1. Combined temporal-and-consumable-resource network, in which intervals for duration (STN) and resource use (cRN) associated with the same activity are linked together. Such links are indicated by dashed lines; the linked intervals are those associated with arcs directed *toward* the nodes of origin and destination for the cross-links. For example, activity x in the cRN is associated with arc (b,d) in the STN.

To support dispatchability, a cRN must allow any sequence of instantiations to be made in the associated STN without exceeding a resource capacity. Given the bijective mapping from STN event to resource use, this implies that any choice of value for an instance of resource use must allow some values to be chosen for all future (as yet uninstantiated) variables. We refer to this loosely as "cRN dispatchability".

The only necessary condition for cRN dispatchability appears to be the following:

$$\sum_{i=1}^{k} lb_i \leq C \tag{1}$$

where lb_i is the lower bound on resource use for the ith activity in a bout. In other words, the sum of the minimum usage for all activities in a bout must not exceed the current capacity.

The strongest sufficient condition is simply that the sum of the upper bounds on resource use be less than or equal to the resource capacity,

$$\sum_{i=1}^{k} ub_i \leq C \tag{2}$$

Obviously, in this case we do not have to consider a cRN at all during execution, since whatever values the Executive selects in the STN, the resulting resource use will be within capacity.

This condition puts potentially severe limitations on the range of choice that can be given to the Executive. This can be seen in Figure 1, where the sum of upper bounds (40) is well above the capacity (30). In order to conform with this condition, the intervals would have to be reduced by one-half on average.

In Figure 1 a constraint is used to insure that resource capacity is not exceeded. In addition, this cRN is fully consistent, because for every value of resource usage chosen for one activity there is at least one usage value for the other that is within the designated bounds.

This consistency can be expressed as:

$$\sum_{i=1}^{k-1} ub_i \; + \; lb_k \; \leq \; C \qquad (3)$$

holding for all subsets of $k-1$ activities in a bout of length k. The question then is, is this condition sufficient to support dispatchability (the second requirement listed above)?

Unfortunately, the answer is, "No". Because of the tight linkage between time and resource use, propagation based on one of these factors will also affect the other. In terms of the dual network shown in Figure 1, either of the following problems may arise.

1. Reductions in cRN upper bounds may delete values in the STN that are necessary to insure dispatchability in the temporal subnetwork.
2. Increasing a lower bound of an STN interval may require an increase in the lower bound of the corresponding resource interval in the cRN, thus violating the cRN dispatchability condition.

For the cRN→STN interaction, if changes occur in the STN, the only part of the graph we have to worry about is between the point of change and variables that are already instantiated (i.e. events that are already fixed), a subset of the uninstantiated nodes that we will call the "critical region". Dispatchability will still hold with respect to future domains by virtue of the original STN dispatchability.

For the STN→cRN interaction, the critical situation involves an increase in the lower bound for an end-time of a resource-activity. However, an increase that would violate the resource condition can only occur if there is an arc to that node from a node other than the start-time. Moreover, a violation can only occur if this additional constraint forces the end-time to be greater than a given value, without putting similar constraints on the start-time. In this case, depending on the start-time chosen, the end-time and hence the activity duration can be forced to take a value greater than the minimum. An example is shown in Figure 2, in which the constraint between b and d could force d to take a time that requires that d-c be equal to 2, as shown by the following sequence of instantiations:

a = 0	b = 1 − 2, c = 9
b = 2	d = 11 − 12
c = 9	d = 11

This situation can be avoided if the end-time is, in fact, independent of the start-time. In this case, when the STN is made dispatchable the additional constraint on the end-time can be replaced with a constraint on the start-time. In Figure 2, this would mean that the constraint between b and d can be replaced with one between b and c, the start-time for the same activity. This can be done (given the triangle inequality) if $|\,\mathrm{bc}\,| \; + \; |\,\mathrm{cd}\,| \; = \; |\,\mathrm{bd}\,|$ [4]. As a result of this manipulation, both the start- and end-times are subject to the same constraint, so the restriction on end-times that we must avoid cannot occur. If this can be done for each such situation involving a critical domain, then the dispatchability guarantee can be established.

This resolution of the STN→cRN problem also solves the cRN→STN problem. Because, if the activity in question has no nodes other than the start-time adjacent to its end-time node, then there will not be any activities in the critical region with domain values that depend on specific values in the domain of its end-time. And this suggests a general statement for a sufficient condition for dispatchability with consumable resources.

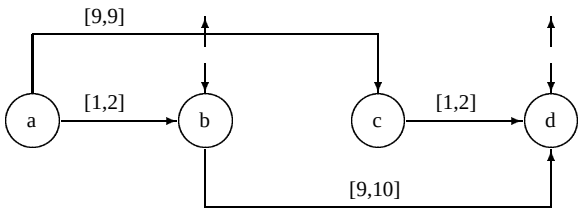

Figure 2. STN component of composite network, in which instantiation of events can lead to violation of required conditions on resource use. Further explanation in text.

4 A General Condition for Dispatchability

The general condition for dispatchability has two parts. First, the following inequality must hold,

$$\sum_{i=1}^{j} ub_i \; + \; \sum_{i=j+1}^{k} lb_i \; \leq \; C \qquad (4)$$

for some subset of $k \; - \; j$ activities. In addition, for each activity associated with a lower bound in the inequation, the end-time must be independent of the start-time. Such activities will be referred to as "buffers", since they serve to buffer the scheduling procedure against going over resource capacity. (The term is due to N. Muscettola.) The second requirement insures that the lower bound values will be available regardless of the values (for times or resource use) chosen for their predecessors and successors.

The object of inequation (4), like that of (3), is to insure that there will be at least one valid assignment in certain domains regardless of the values chosen for assignment from the others. The difference is that here we are specifying a particular set of activities. This reduces flexibility of execution in comparison with the former condition. However, in many cases it should be possible to ameliorate this situation, as indicated below.

In addition, the present condition will not allow us to use n-ary resource constraints of the form shown in Figure 1 except under special circumstances, one of which was described previously [6]. In fact, at first blush it might appear that the present condition amounts to a special case of inequation (2), with some upper bounds set equal to the lower bounds for the same activities. However, this is not really the case, as we demonstrate in the next section.

5 Strategies for Flexible Resource Use

First, it should be noted that, if inequality (4) holds for two sets of buffers within a bout, and one set is a subset of the other, then the smaller set can be used. More generally, the buffer-sets form a lattice structure, and we can use one of the minimal sets to satisfy the condition. Ideally, this would be a singleton. Moreover, if there is more than one such singleton set, then these can be tracked during execution, so that buffer-activities instantiated before the last one do not have to be set to their lower bounds. (One could track in this way with sets of larger size, but one could not avoid using lower bounds until one came to the last member of a given set.)

Although, in general, ordinary n-ary resource constraints cannot be used in connection with this condition, another strategy described

in previous work [6] is still viable. The idea is to track the difference between the upper bound on resource use for a given activity and the actual use. The accumulated difference between these values,

$$\sum_{i=1}^{j} ub_i \; - \; \sum_{i=1}^{j} r_i, \tag{5}$$

which we call the "credit", can be added to the lower bound of a buffer-activity to increase the range of values consistent with the requirements for dispatchability. If, in addition, it is possible during planning to make adjustments to schedule times to insure that a minimal buffer set occurs as late as possible during a bout of activities, then we can increase flexibility further, since there is more opportunity for credit to accumulate.

Carrying the latter idea further, if we can insure that all of the activities that might occur last (the "candidate-last" set) are buffers, then we *can* use ordinary n-ary resource constraints during execution. In this case, the constraint will insure that the last activity is set to a value that satisfies the dispatchability requirements (and the fact that it is a buffer will insure that acceptable values are present).

Finally, if there are singleton buffers, it may be worthwhile to try to satisfy inequality (3) for all subsets of $k - 1$ activities. In this case, if a value for resource use is chosen for any activity that is no more than its lower bound plus the existing credit, then the Executive will be able to choose any possible value for the remaining resource activities.

Thus there are a variety of strategies that can be used in a dispatchable execution with consumable resource constraints. Selection of an appropriate strategy depends on the character of the dispatchability condition. If inequality (2) is satisfied, then a simple STN is sufficient. If the condition associated with inequality (4) is satisfied, then a minimal buffer-set can be used, together with credit-based strategies. Finally, there are cases where a strategy involving ordinary n-ary resource constraints is sufficient.

6 Conclusions

In this work we have derived conditions that insure that schedule execution is dispatchable even when consumable resources are involved. This involved pinpointing the basic difficulties that arise when combining temporal and consumable resource constraints, which take the form of interactions between two subnetworks. We have also shown that this can be done while still allowing, in many cases, a high degree of flexibility in selecting values for resource use during execution.

The present work is pertinent to a large class of problems encountered in planning by autonomous systems like the Remote Agent, when consumable resources like the Solid State Recorder are involved. In these systems, greater flexibility of resource use can make operations involving tasks such as data collection more effective, thus increasing the likelihood that overall mission goals are accomplished.

7 Acknowledgements

This work was carried out in association with the NASA-Ames Reseach Center, Moffet Field, CA and was partly supported by the National Science Foundation under Grant No. IRI-9504316. The paper benefited from several discussions with P. Morris and N. Muscettola.

REFERENCES

[1] R. Dechter, I. Meiri, and J. Pearl, 'Temporal constraint networks', *Artificial Intelligence*, **49**, 61–95, (1991).

[2] P. Jeavons, D. Cohen, and M. Gyssens, 'A unifying framework for tractable constraints', in *Principles and Practice of Constraint Programming - CP '95*, eds., U. Montanari and F. Rossi, volume 976 of *Lecture Notes in Computer Science*, 276–291, Springer, Berlin, (1995).

[3] N. Muscettola, 'HSTS: Integrating planning and scheduling', in *Intelligent Scheduling*, eds., M. Zweben and M. S. Fox, 169–212, Morgan Kaufmann, San Mateo, CA, (1994).

[4] N. Muscettola, P. Morris, and I. Tsamardinos, 'Reformulating temporal plans for efficient execution', in *Proceedings KR-98*, (1998).

[5] N. Muscettola, P. Pandurang Nayak, B. Pell, and B. C. Williams, 'Remote agent: to boldly go where no AI system has gone before', *Artificial Intelligence*, **103**, 5–47, (1998).

[6] R. J. Wallace and E. C. Freuder, 'Dispatchable execution of schedules involving consumable resources', in *Proceedings, Fifth International Conference on Artificial Intelligence Planning and Scheduling (AIPS-2000)*, pp. 283–290, Menlo Park, CA, (2000). AAAI Press.