# COMMENT

## Merging the Semantics of State Constraints for Collections of Activities

**Miriam H. Nadel**
The Aerospace Corporation
P. O. Box 92957
Los Angeles, CA, 90009-2957
miriam.nadel@aero.org

### Introduction

Coupling between activities can create a number of practical problems for scheduling. Aggregation of activities and related constraints can reduce complexity, allowing greater computational efficiency in scheduling. This commentary is divided into

### Comment on Motivation

Interactions between constraints on shared states and resources are clearly a significant issue for scheduling. One key point about scheduling activities individually is that scheduling decisions are not commutative. If you determine that you can't schedule a particular activity, you need to consider whether or not to reexamine the activities that are already on the schedule. Attempting to reschedule existing activities quickly causes one to run into the "curse of dimensionality" and processing time often becomes a practical limitation. However, if you don't attempt rescheduling, underutilization of assets is likely. It is only when all of the activities and constraints are viewed simultaneously, that one can identify all of the scheduling options. Therefore, merging activities effectively reduces the dimensionality of the scheduling problem.

Another consideration is that some types of constraints can be recast in the form of additional activities. Use of antennas for satellite ranging, for example, typically requires a number of setup and configuration activities for the antenna, in addition to the actual ranging data collection. Typically, those activities are just lumped with the collection request. This is inefficient, as it doesn't allow insight into situations in which multiple requests from different users require the same antenna configuration. By treating the configuration as a separate activity, which interacts with the collection activity, it may be possible to achieve more efficient schedules.

A final practical motivation arises in a multi-user environment, where political considerations can affect the scheduling process. Prioritization of tasks (done by some external agency, rather than the developers of the scheduling system) is typical in such situations. Being able to aggregate activities could eliminate some of the flak when a lower-priority activity gets scheduled because it has a positive interaction with a higher-priority activity, while some activity with intermediate priority is bumped.

### Modeling

The essence of the approach in this paper is the modeling of activities as activities in terms of reservations and the transformation of a set of individual reservations into a related set of non-interacting reservations. The wide range of domain examples is interesting and relevant to many other space-related scheduling applications. The effort involved in translating a domain into an expression of a model could be considerable. In practice, translating scheduling problems into any modeling language is often one of the hardest aspects of utilizing existing algorithms and has been one of the reasons for the limited application of several commercial software packages.

### Solution Methodology

The concept of transforming the set of interacting reservations, P, into a set of non-interacting reservations, P' appears to be a powerful method. One area that isn't addressed explicitly, but is of considerable practical importance, is conflict resolution. A number of situations are discussed in which no possible non-conflicting set of intervals exists. This information is often useful to users, who may then redefine their requested activities.

### Quality Measures - Accuracy

Random placement is effectively a worst case bound on performance (and a best case bound for flexibility). It would be interesting to call out false positives vs. false negatives when assessing accuracy. Depending on the exact problem, one may be more significant than the other. For example, if you have an undersubscribed system, false negatives may not be a major concern, although they're undesirable from the standpoint of producing an excessively conservative solution. It's harder to think of a situation in which false positives would be insignificant, but that might be the case where some constraint can be violated temporarily. For example, a sensor tasking system may allow violation of a power constraint for a brief period of time.

## Quality Measures - Speed

The difference in computational time for the informed and the naïve approaches is considerably different for the domains discussed. The number of activities involved (or, in particular, the activity group size) is clearly one factor in how long it takes to solve a given problem, but this is true for both scheduling algorithms. How many interactions there are between activities is probably predictive of how much longer the informed approach will take for a given algorithm, but this is not necessarily obvious a priori for a given domain.

Of course, computational load is more significant for some problems than for others. A very stressing example is near-real-time tasking of one sensor in response to an observation by another sensor. That case requires detecting an event, recomputing the schedule, calculating commands to slew the sensor to the new location, and transmitting commands to the satellite (possibly via ground relays, depending on satellite location) in a very short amount of time. Similarly stressing timelines arise in other domains, e.g. weapon to target assignment for ballistic missile defense.

### Efficacy Assessment

Efficacy assessment is a broader issue than just effectiveness in conflict resolution and the number of problems solved. Because the problems used for the empirical evaluation here were generated randomly, it's not easy to place bounds on how many problems could be solved. That is, we don't know how many solvable problems the informed search method was still unable to solve. This is important because a lot of real world problems are unlikely to solvable. An oversubscribed system can never schedule all of the requested activities and, therefore, other metrics may be more appropriate in those circumstances.

## Conclusion

The background of this commenter includes sensor tasking for surveillance applications, as well as scheduling of antenna resources for satellite control applications. The focus of the commentary has, therefore, been on considerations for practical applicability to problems that arise in implementing schedulers in these domains. The commenter hopes that the issues raised in this paper will highlight pertinent issues in broadening the application of the methodology.

## Questions?

Contact Miriam Nadel (miriam.nadel@aero.org)