Scheduling with uncertainty: A proactive approach using Partial Order Schedules

Article *in* Ai Communications · January 2005

Source: DBLP

1 author:

Nicola Policella
European Space Agency

AI Communications 18 (2005) 165–167
IOS Press

Thesis

Scheduling with uncertainty: A proactive approach using Partial Order Schedules

Nicola Policella
DIS, Computer and System Science Department, University of Rome "La Sapienza", Italy
E-mail: policella@dis.uniroma.it

This work addresses the broad question of how to build schedules that are robust in the face of dynamic execution environments. This question is of considerable practical importance, as a major obstacle to the use of schedules in practice is their brittleness when activities cannot be executed as planned.

Scheduling can be defined as the assignment of start and end times to a set of activities (or tasks) subject to a number of constraints. Constraints are typically recognized in either time constraints and/or resource constraints. Task synchronization in the production chain of a factory, management of space missions, and transportation scheduling to support crisis management, are only but a few representative examples. However, the synthesis of initially feasible schedules is hardly ever sufficient as, in real-world working environments, unforeseen events tend to quickly invalidate the schedules predictive assumptions and bring into question the continuing validity of the schedule's prescribed actions. Therefore, to safeguard against unforeseen events, the thesis defended in this dissertation is that: *flexible solutions, and in particular Partial Order Schedules, are able to better absorb unpredicted or exogenous events in terms of providing a rapid answer to the need of a new solution.*

The main limitation in the application of most current scheduling approaches lies in the generation of fixed-time solutions, where at each activity is associated a fixed starting time. Such solutions may exhibit a high degree of brittleness. In fact any external change rapidly invalidate the schedule's predictive assumptions and bring into question the continuing validity of the schedule's prescribed actions.

The alternative approach pursued in this work consists of adopting a graph formulation of the scheduling problem, wherein activities competing for the same resources are simply ordered to establish resource feasibility, and it is possible to produce schedules that retain temporal flexibility where allowed by the problem constraints (Fig. 1). In essence, such a "flexible schedule" encapsulates a set of possible fixed-time schedules, and hence is equipped to accommodate some amount of the uncertainty at execution time. Following this intuition we have introduced the definition of *Partial Order Schedules*, or $\mathcal{POS}$s [6]. A $\mathcal{POS}$ consists of a set of feasible solutions for the scheduling problem that can be represented in a compact way by a temporal graph, that is, a graph in which any activity is associated to a node and temporal constraints define the order in which such activities have to be executed.

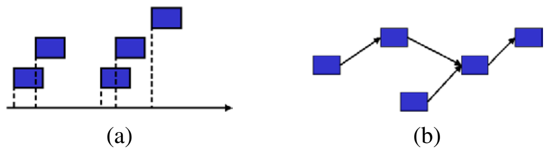

Fig. 1. Partial order schedules vs fixed-time solutions. (a) Fixed-time schedule: each activity is anchored to a precise start time; (b) Partial Order Schedule: each activity is free to be allocated in the interval defined by the temporal graph.

To provide a more formal definition of a $\mathcal{POS}$ we use the activity on the node representation: given a problem P, this can be represented by a graph $G_P(V_P, E_P)$, where the set of nodes $V_P = V \cup \{a_0, a_{n+1}\}$ consists of the set of activities specified in P and two dummy activities representing the origin (a_0) and the horizon (a_{n+1}) of the schedule, and the set of edges E_P contains P's temporal constraints between pairs of activities. A solution of the scheduling problem can be represented as an extension of G_P, where a set E_R of simple precedence constraints, $a_i \prec a_j$, is added to remove all the possible resource conflicts. Given these concepts, a *Partial Order Schedule* is defined as follows:

Definition 1. Given a scheduling problem P and the associated graph $G_P(V_P, E_P)$ that represents P, a *Partial Order Schedule*, $\mathcal{POS}$, is a set of solutions that can be represented by a graph $G_{POS}(V_P, E_P \cup E_R)$.

It is worth noting that a partial order schedule provides an immediate opportunity to reactively respond to some of the possible external changes by simply propagating their effects over the "graph", by using a polynomial time computation [4]. In fact the augmented duration of an activity, as well as a greater release time, can be modeled as a new temporal constraint to post on the graph. It is also important to note that, even though the propagation process does not consider the consistency with respect the resource constraints, it is guaranteed to obtain a feasible solution by definition of $\mathcal{POS}$s. Therefore a partial order schedule provides a means to find a new solution and ensures its fast computation.

In addition to the definition of partial order schedule, in this work we have analyzed two different, somehow complementary, methodologies to generate this kind of scheduling solutions. In fact, an important open question is how to generate flexible schedules with good robustness properties.

The methodologies implemented are both based on the Constraint Satisfaction Problem paradigm [1,7]. The ability of constraints to capture arbitrary relations makes them a natural modeling paradigm for our purposes. In fact, it is possible to easily represent both mathematical or logical formulae and arbitrary relations with constraints. Therefore, the constraint programming approach satisfies the necessity of being able to represent the different aspects of the problem and retain the capability of guiding the search exploiting the knowledge of the problem. More precisely, we use a particular CSP formulation of a scheduling problem: the Precedence Constraint Posting model [3]. This approach consists of posting the set of additional precedence constraints that are needed among the sets of activities that are competing for the same resources. One principal advantage of this sequencing approach is that it avoids over-commitment, as activities need not be anchored to specific start times during the search or in the final solution.

In this dissertation, two methods have been described. First, a least commitment approach that uses computed bounds [5] on cumulative resource usage to identify potential resource conflicts, and progressively winnows the total set of temporally feasible solutions into a smaller set of resource feasible solutions by resolving detected conflicts.

A different, less intuitive, approach to $\mathcal{POS}$ synthesis is based on a phase separation and called *Solve-and-Robustify*: under this schema, a feasible fixed-times schedule is first generated in stage one, and then, in the second stage, the initial solution is transformed into a temporally flexible schedule. In this second step, fixed-time commitments are converted by a heuristic technique, called *chaining*, into a sequences (chains) of activities to be executed by various resources.

A chaining procedure transforms a feasible fixed-time solution into a $\mathcal{POS}$ by dispatching activities to specific resource units. Since choices can be made as to how to dispatch activities to resource units, it is possible to generate different $\mathcal{POS}$s from the same starting solution, and these different $\mathcal{POS}$s can be expected to have different robustness properties. For this reason, iterative sampling is used to explore this set of solutions. Randomization is added to obtain a different solution at each iteration and, in so doing, to generate a sequence of $\mathcal{POS}$s starting from the same initial schedule. Additionally, different heuristics have been explored that change the effectiveness of chaining. A relevant aspect of the chaining procedure is its ability of preserving different properties of the initial, fixed-time solutions like the makespan and the lateness. In fact, the original solution is always included in the generated $\mathcal{POS}$.

In conclusion, in the thesis we introduce the definition of partial order schedule as a particular set of schedules and we explicitly consider the problem of how to take advantage of temporal flexibility to hedge against the dynamics of execution. We produce several new results and insights into the issue of how to build partial order schedules. In this regard two broad classes of constraint based scheduling procedure are defined and investigated: least commitment engines attempt to generate as few (and as little constraining) delay constraints as possible, in order to solve potential resource conflicts and guarantee that all the schedules consistent with the generated delay constraints are feasible; by contrast, Solve-and-Robustify approaches generate first one "good" schedule, and then derive a set of delay constraints from this schedule. Surprisingly, these Solve-and-Robustify approaches have been found to dominate the least commitment ones in their ability to produce schedules with better robustness properties (as measured in several different ways) on a set of benchmark resource-constrained project scheduling problems from the Operations Research literature [2]. Additionally, the Solve-and-Robustify approaches have been found to solve greater numbers of problem instances, produce better makespan results, and solve problems in significantly less computation time.

References

[1] P. Baptiste, C. Le Pape and W. Nuijten, *Constraint-Based Scheduling*, Vol. 39 of *International Series in Operations Research and Management Science*, Kluwer Academic Publishers, 2001.

[2] M. Bartusch, R.H. Mohring and F.J. Radermacher, Scheduling project networks with resource constraints and time windows, *Annals of Operations Research* **16** (1988), 201–240.

[3] A. Cesta, A. Oddi and S.F. Smith, A Constraint-based method for Project Scheduling with Time Windows, *Journal of Heuristics* **8**(1) (2002), 109–136.

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

[5] N. Muscettola, Computing the envelope for stepwise-constant resource allocations, in: *Principles and Practice of Constraint Programming, 8th International Conference, CP 2002*, Vol. 2470 of *Lecture Notes in Computer Science*, Springer, 2002, pp. 139–154.

[6] N. Policella, Scheduling with uncertainty: A proactive approach using Partial Order Schedules, PhD thesis, University of Rome "La Sapienza", March 2005.

[7] E.P.K. Tsang, *Foundations of Constraint Satisfaction*, Academic Press, London and San Diego, 1993.