Available online at www.sciencedirect.com
ScienceDirect
Discrete Optimization 4 (2007) 288–314
DISCRETE OPTIMIZATION
www.elsevier.com/locate/disopt

Strong polynomiality of resource constraint propagation

Luc Mercier, Pascal Van Hentenryck*

Brown University, Box 1910, Providence, RI 02912, United States

Received 14 October 2005; received in revised form 18 May 2006; accepted 10 January 2007
Available online 29 October 2007

Abstract

Constraint-based schedulers have been widely successful in tackling complex, disjunctive, and cumulative scheduling applications by combining tree search and constraint propagation. The constraint-propagation step is a fixpoint algorithm that applies pruning operators to tighten the release and due dates of activities using precedence or resource constraints. A variety of pruning operators for resource constraints have been proposed; they are based on edge finding or energetic reasoning and handle a single resource.

Complexity results in this area are only available for a single application of these pruning operators, which is problematic for at least two reasons. On the one hand, the operators are not idempotent, so a single application is rarely sufficient. On the other hand, the operators are not used in isolation but interact with each other. Existing results thus provide a very partial picture of the complexity of propagating resource constraints in constraint-based scheduling.

This paper aims at addressing these limitations. It studies the complexity of applying pruning operators for resource constraints to a fixpoint. In particular, it shows that: (1) the fixpoint of the edge finder for both release and due dates can be reached in strongly polynomial time for disjunctive scheduling; (2) the fixpoint can be reached in strongly polynomial time for updating the release dates or the due dates but not both for the cumulative scheduling; and (3) the fixpoint of "reasonable" energetic operators cannot be reached in strongly polynomial time, even for disjunctive scheduling and even when only the release dates or the due dates are considered.

Keywords: Constraint-based scheduling; Resource constraint; Complexity; Edge finder; Energetic reasoning

1. Introduction

Constraint-based schedulers (e.g., [1,2,8–10]) are widely successful in tackling complex, disjunctive, or cumulative scheduling problems in manufacturing, transportation, supply-chain management, and the steel industry. These problems often consist of minimizing the completion time of a set of jobs, each job being a sequence of tasks linked by precedence constraints. Each task has a processing time and may require some units of one or more resources. Disjunctive resources have a capacity of 1, and two tasks requiring the same disjunctive resource cannot overlap in time. Cumulative resources have a finite capacity C, and the total demand for a cumulative resource at any time t cannot exceed C.

* Corresponding address: Brown University, Department of Computer Science, Box 1910 Providence, RI 02912, United States.
E-mail addresses: mercier@cs.brown.edu (L. Mercier), pvh@cs.brown.edu (P. Van Hentenryck).

1572-5286/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.disopt.2007.01.001

Constraint-based schedulers approach the solving of complex scheduling problems by iterating two main steps until a feasible or optimal solution is found:

1. a constraint propagation step that tightens the release and due dates of each activity;
2. a (nondeterministic) branching step that adds new precedence constraints or assigns a starting date to some activity.

This paper focuses on the constraint propagation step that tightens the release and due dates of each activity. The propagation algorithm applies pruning operators until a fixpoint is reached, i.e., until no further tightenings of the release and due dates is possible. When operators are monotone, a property most of them satisfy, this process converges to a unique fixpoint independently of the application order of the operators.

Because even the one-resource problems are NP-complete in the presence of release and due dates, these pruning operators apply tightening rules exploiting necessary conditions for feasibility. Much research was devoted to the definition of such tightening rules and their associated pruning operators, trying to achieve a good compromise between their runtimes and the quality of their pruning. On the quality side, Baptiste et al. [2] provide an impressive collection of dominance properties on a set of operators; such a set dominates another one if its constraint propagation converges to a smaller (inclusively speaking) fixpoint. Unfortunately, on the runtime side, the performance of these operators is poorly understood at this point. Typically the complexity of a single operator application is known but the complexity of reaching the fixpoint is not addressed. More precisely, the following issues should be addressed:

- How many operator applications are necessary to reach a fixpoint?
- Is the complexity independent of the order in which the operators are applied?
- What is the complexity of computing the fixpoint for the composition of operators?

This paper is, to our knowledge, the first attempt to study these questions. Although some authors, including Torres and Lopez [15], Vilim et al. [17], and Péridy and Rivreau [14,13], took the propagation process into account when designing pruning operators, they did not directly study the questions above (see Section 7 for a review of these interesting results).

This paper first proposes a general framework, *propagation patterns*, to study these issues formally. It presents some general properties of propagation patterns, allowing the proofs of subsequent results to be both simpler and more general. The paper then applies the framework to the study of two classes of popular operators: edge finding and energic reasoning. Its main results are:

- For disjunctive resources, the fixpoint of the edge finder for both release and due dates can be reached in strongly polynomial time;
- For the cumulative resources, the fixpoint can be reached in strongly polynomial time for updating the release dates or the due dates but not both;
- The fixpoint of "reasonable" energetic operators cannot be reached in polynomial time, even for disjunctive scheduling and even when only the release dates or the due dates are considered.

These results offer a fundamentally different picture of the computational complexity of these pruning operators. In particular, they identify fundamental differences in efficiency when these operators are iterated in fixpoint algorithms. The proofs also indicate some "flaws" in existing definitions of energetic operators; they also suggest some directions in order to address their pathologic behaviors in fixpoint computations, especially for cumulative scheduling. Finally, the results formally demonstrate the significance of the algorithms of Péridy and Rivreau, at least from a complexity standpoint.

The rest of the paper is organized as follows. Section 2 reviews the technical background behind this paper. Section 3 introduces the concept of propagation patterns to formalize constraint propagation algorithms, which may differ in the order in which they apply to the pruning operators. Section 4 proves some fundamental results on propagation patterns. In particular, it shows that one can essentially focus on a specific class of propagation patterns to derive the complexity results. Section 5 presents the complexity results for disjunctive resources, while Section 6 presents those for cumulative results. Section 7 reviews some related work, and discusses previous results in light of our new work. Section 8 concludes the paper and presents the open issues. The proofs not given in the paper are in the appendices.