Available online at www.sciencedirect.com
ScienceDirect
Computers and Chemical Engineering 32 (2008) 275–291
Computers & Chemical Engineering
www.elsevier.com/locate/compchemeng

Scheduling of multi-product batch plants based upon timed automata models

Sebastian Panek[a,*], Sebastian Engell[a], Subanatarajan Subbiah[a], Olaf Stursberg[b]

[a] *Process Control Lab (BCI-AST), Department of Biochemical and Chemical Engineering, Universität Dortmund, 44221 Dortmund, Germany*
[b] *Institute of Automatic Control Engineering, Technische Universität München, 80290 München, Germany*

Received 1 November 2006; received in revised form 17 July 2007; accepted 18 July 2007
Available online 2 August 2007

Abstract

This paper reports on the successful application of a recently proposed method for solving scheduling problems by reachability analysis of timed automata. The jobs and the resources are modeled by a set of synchronized timed automata, and the scheduling problem is solved by a cost-optimal symbolic reachability analysis. The analysis searches for a path of states and state transitions from an initial state to a terminal state in which all jobs have been finished. The appeal of the approach is the intuitive and modular graphical modeling, and an efficient solution, as is illustrated here for a well-known case study from the chemical industry.
© 2007 Elsevier Ltd. All rights reserved.

Keywords: Multi-product batch plants; Scheduling; Timed automata; Reachability analysis

1. Introduction

1.1. Standard approaches

Optimal scheduling of multi-product batch plants (resource allocation, sequencing and timing) contributes significantly to the productivity of these plants and has been in the focus of research for many years now. The standard approach to this class of problems is to model the problem at hand by a state-task network (STN) (Kondili, Pantelides, & Sargent, 1993; Shah, Pantelides, & Sargent, 1993) or by a resource task network (RTN) (Pantelides, 1994) and to transform the graphical model together with additional information into (usually linear) mathematical programs with integer variables (MILP) that are then solved by efficient standard solvers for this class of problems as, e.g. CPLEX (ILOG, 2002). The efficiency of the solution of the optimization problems strongly depends on the formulation of the mathematical model, equivalent problem descriptions may result in large differences in the performance. In particular, the representation of time has been discussed in many papers: formulations with time slots, global events and unit-specific events emerged in the past decade and were thoroughly studied and compared (Castro, Barbosa-Povoa, Matos, & Novais, 2004; Ierapetritou & Floudas, 1998a; Ierapetritou & Floudas, 1998b; Ierapetritou, Hene, & Floudas, 1999; Janak, Lin, & Floudas, 2004; Maravelias & Grossmann, 2003; Mendez & Cerda, 2000; Shaik, Janak, & Floudas, 2006; Sundaramoorthy & Karimi, 2005; Wang & Guignard, 2002). The formulation of mathematical programs for batch scheduling requires deep insight into the interaction of the structure of the model with the strategy of the solver, nonetheless, the solver performance is hard to predict, small changes in the model may lead to large variations of the time required for its solution (Till, Engell, Panek, & Stursberg, 2004). And despite the significant advances in this field, the computation times required to obtain a first feasible solution, a near-optimal solution (with a small optimality gap) or the true optimum are still large, often prohibitively large especially when applications in which a reaction to events within a limited amount of time is required (online or reactive scheduling).

1.2. Scheduling with timed automata

A promising alternative that has been discussed recently is to model scheduling problems by timed automata (TA) and to solve these problems by *reachability analysis*. Numerical

[*] Corresponding author.
E-mail addresses: s.panek@bci.uni-dortmund.de (S. Panek), s.engell@bci.uni-dortmund.de (S. Engell), s.subbiah@bci.uni-dortmund.de (S. Subbiah), stursberg@tum.de (O. Stursberg).

0098-1354/$ – see front matter © 2007 Elsevier Ltd. All rights reserved.
doi:10.1016/j.compchemeng.2007.07.009

experiments for hard job-shop benchmark examples showed that the performance of this approach is comparable or superior to other approaches (Abdeddaim & Maler, 2001; Panek, Stursberg, & Engell, 2004, 2006). Moreover, the modeling is completely graphical and the problem can be represented in a modular and hence transparent fashion with independent small models for recipes, resources, etc. that are connected by synchronization labels and shared variables and can be composed automatically.

Timed automata were proposed in (Alur & Dill, 1994) as an extension of finite automata by clocks in order to model problems with timing constraints. The original purpose of TA was to model and to analyze properties of timed systems, such as real-time software, embedded systems, controllers, and network protocols. The algorithmic verification of TA models comprises reachability analysis to determine the reachable subset of states, and the evaluation whether formal properties (as safety or liveness) are satisfied for the reachable states. Quite efficient and user-friendly tools as, e.g. Uppaal (Behrmann, Larsen, & Rasmussen, 2005), Kronos (Yovine, 1997) or IF (Mounier, Graf, & Bozga, 2002) have been developed for this purpose.

Recent research revealed that reachability analysis of TA is also suitable for optimization and scheduling (Abdeddaim & Maler, 2001; Behrmann, Brinksma, Hendriks, & Mader, 2005; Behrmann, Larsen, et al., 2005; Fehnker, 1999; Rasmussen, Larsen, & Subramani, 2004). In the context of scheduling, the objective is to explore the *reachability graph* (a representation of the state space) and to find a path from an initial state to a target state in which all operations are finished while satisfying all timing requirements. If such a path is found and fulfills an optimization criterion, the solution to the scheduling problem is obtained.

In the *priced timed automata* (PTA) modeling scheme, originally introduced in (Alur, La Torre, & Pappas, 2001; Behrmann et al., 2001), transitions and staying in locations cause costs, such that the overall cost of a path (or schedule) is equal to the sum of the transition costs and the integral costs for the residence in states. Cost-optimal reachability analysis for PTA aims at finding the cost-optimal path from an initial state to a target state. Specialized software tools like Uppaal-CORA (Behrmann, Larsen, et al., 2005) and TAopt (Panek et al., 2004, 2006) were developed for cost-optimal reachability analysis of priced TA. Extensions to problems with uncertainty are discussed in (Abdeddaim, Asarin, & Maler, 2006). Despite the recent advances in this area, the existing contributions lack a systematic approach to industrial scale problems with complex constraints beyond the limited problem class of pure job-shops. In this paper, we give a comprehensive introduction into the modeling of scheduling problems by timed automata and in the techniques to solve the resulting optimization problems by reachability analysis. For an efficient solution, state space reduction techniques must be used. Some of these techniques are "safe", i.e. they do never prune the optimal solutions while others are more efficient but may lead to suboptimal solutions. A significant limitation of the approach is that only problems of timing in the broad sense (resource allocation, sequencing and timing of operations) can be addressed, but not, e.g. batch sizing whereas MILP approaches can tackle both simultaneously. These must be tackled by an algorithm that uses the solution of the scheduling problems to determine an optimal solution, e.g. by meta-heuristics.

The main contribution is the extension and the application of state space reduction techniques to TA models of industrial scheduling problems. In particular, methods such as *non-laziness* and the *sleep-set method* described in (Abdeddaim & Maler, 2001; Godefroid, 1991; Panek et al., 2006) for job-shops are extended to problem classes which comprise more complex constraints. These extensions push the scope of the TA approach towards real-world problem scenarios.

The remainder of this paper is organized as follows: Section 2 briefly explains the syntax of TA. Section 3 deals with job-shop scheduling and the question how to formulate modular TA models for job-shops. Section 4 answers the question how modular models can be composed. Section 5 explains the semantics and abstraction techniques of TA. Section 6 presents the basic algorithm for the reachability analysis of TA. Section 7 discusses the assumption of immediate runs and its consequences for symbolic transitions and the symbolic reachability graph. Section 8 discusses various state space reduction techniques which aim at reducing the reachability graph and thus the search effort. Section 9 broadens the scope of job-shops to more complex industrial problems. Advanced state space reduction techniques are described in Section 10. Finally, Section 11 presents the results of computational studies for a variant of a well-known benchmark problem from the chemical industry. Conclusions and outlook close this paper in Section 12.

2. Timed automata

Timed automata (TA) are discrete automata extended by the concept of clocks. Clocks are special variables c the values of which increase with the rate $\dot{c} = 1$. Clocks in TA can be reset to 0 but not stopped – after a reset $c := 0$ the value of a clock continues increasing. Clock conditions can be evaluated to enable transitions (transition guards) and to check staying conditions in locations (invariants). The states of TA are composed of discrete locations (like in discrete automata) and vectors of clock values. Discrete transitions in TA switch from one location to a neighbor location while time transitions wait in a location and increase the values of all clocks by the same period.

We only give a short and informal definition of timed automata here, and refer to (Behrmann et al., 2001) for complete definitions and for priced timed automata (PTA). A timed automaton is a tuple (L, l_0, C, I, T) in which: L is a set of locations with an initial location $l_0 \in L$, and C is a set of clocks; I is a set of invariants, i.e. conjunctions of simple constraints $c \leq k$, where $c \in C$ and $k \in \mathbb{R}^{\geq 0}$. A location can only be active if the invariant is true. T is a set of transitions (l, g, a, r, l'), each of which leads from a location $l \in L$ to another location $l' \in L$; the guard g is a conjunction of constraints $c \sim k$ or $c - c' \sim k$ with c, $c' \in C$, $k \in \mathbb{R}^{\geq 0}$, and $\sim \in \{==, \neq, <, >, \leq, \geq\}$; $a \in Act$ is an action label and $r \subset C$ is a set of clocks to be reset after taking the transition. A transition can be only executed if the guard evaluates to true.