Modelling: Scheduling and Planning

University of Dortmund

3rd May 2004

AMETIST DELIVERABLE 1.4

Project acronym: AMETIST
Project full title: Advanced Methods for Timed Systems
Project no.: IST-2001-35304 Project Co-ordinator: Frits Vaandrager
Project Start Date: 1 April 02
Duration: 36 months
Project home page: http://ametist.cs.utwente.nl/

1 Modelling of Job-Shop Scheduling Problems with Timed Automata

In [1, 4, 5] a general TA-model for non-preemptive job shop problems is described. This modelling approach has been successfully applied to benchmark problems up to medium size job shop problems as well as to the Axxom case study with 29 jobs. For makespan minimization or for the generation of schedules that meet all deadlines, the exclusion of lazy schedules is a crucial step towards an effective solution. In the TA approach, this can be included in the model formulation, thus the search space is much reduced compared to a mathematical programming formulation which includes all possible timings. Near optimal solutions can be computed by guided search in computation times which are competitive with other approaches. The approach has been extended to preemptive scheduling problems [2]. These can be modeled as shortest path problems defined on stopwatch automata, an extension of timed automata where some of the clocks can be frozen at certain states. Although standard verification problems on stopwatch automata are known to be undecidable, it could be shown that due to well-known properties of optimal schedules, the shortest path in the automaton belongs to a finite class of acyclic paths where transitions occur at integer points in time, and hence the problem is solvable. Several algorithms and heuristics for finding the shortest paths in such automata were proposed and tested on benchmark examples, giving competitive performance.

A more general model with respect to the cost function are priced timed automata where staying in a state and taking a transition is coupled to a (non-negative) increment of a cost function. In [17] the first steps in modelling the *Axxom* value chain management problem with priced timed automata are reported. The focus is on a clear derivation of a timed automaton model that makes the design steps explicit.

[10] deals with optimal scheduling of acyclic branching programs on parallel machines. The goal is to find the worst-case optimal schedule of a program with if-then-else clauses on several processors. This is a problem of scheduling under uncertainty because the results of the statements are not known in advance and it cannot be solved in a satisfactory manner using static or fixed priority scheduling. Timed automata technology is used to derive scheduling strategies using algorithms for finding shortest paths on game graphs. [5] develops a methodology for treating the problem of scheduling partially-ordered tasks on parallel machines. It is shown how release times and deadlines can be easily incorporated into the model.

For a wider application of TA tools to the modelling and solution of planning and scheduling problems, providing automatic translations of standard problem representations to TAs is needed. Tools like UPPAAL offer a very rich modelling environment which on the other hand is not easily understood by users without a background in timed automata. [11] describes how UPPAAL models can be generated automatically from PDDL3 problem descriptions. PDDL3 is a standard language for the definition of planning problems. In Dortmund, a prototype tool which generates both TA models and MILPs [14] from a standard representation of job shop problems is under development.

2 Online (real time) scheduling and scheduling under uncertainty

Real scheduling problems are characterized by uncertainty about the future evolution of the problem at hand, both with respect to demands and to resources. Thus in sharp contrast to academic static job shop problems where a problem definition is given once and for all, in reality schedules must be computed in a causal information structure in which new information on the actual development of the situation is obtained continuously and the scheduler has to react to this situation. This has been tackled in the work within and related to AMETIST in the groups involved by two different approaches. In [3, 1] it is proposed to synthesize scheduling laws (state dependent decision functions) based upon the assumption of a certain scenario for the future (the not yet fixed part of the schedule). The scheduler then reacts to the actual evolution of the system by applying this fixed strategy, not a fixed schedule, similar to a feedback law in control. For prototype problems, it could be shown that this strategy is superior to simple reactive strategies as e.g. holes filling. However, the computational effort thus far restricts this approach to small problems (e.g. 4 jobs, 6 machines). Also, probabilistic formulations with exponential distributions of the operating times were considered.

The second approach is based on re-scheduling at certain instances with a scenario-based description of the possible future evolutions, taking the possible recourse actions (the future reaction to the realization of chance) into account. [13, 19, 12]. The optimization criterion is the expected cost (or profit) over all scenarios of the future. It could be demonstrated that problems with 1000 scenarios can be solved using a tailored MILP approach.

For a valid comparison of scheduling techniques, pre-computed (robust) schedules or reactive schedulers must be tested for a large number of realizations of chance. The MoDeST modeling language pairs modeling features from stochastic process algebra and from timed and probabilistic automata with light-weight notations such as exception handling. It is supported by the Motor tool, which facilitates the execution and evaluation of MoDeST specifications by means of the discrete event simulation engine of the Moebius tool. [6] describes the application of MoDeST Motor and Moebius to the Axxom case study. Different schedules generated by UPPAAL are embedded in a MoDeST failure model of the lacquer production line, and analyzed with the discrete event simulator of Moebius. This provides insight into the quality of the schedules with respect to timeliness, utilization of resources, and sensitivity to different assumptions about the reliability of the production line.

3 Nonstandard scheduling problems

Optimization of timing behaviour of manufacturing systems can be regarded as a scheduling problem in which tasks model the various production processes. Typical for many manufacturing systems is that (collections of) tasks can be associated with manufacturing entities, which can be structured hierarchically. Execution of production processes for several instances of these entities results in nested finite repetitions, which blows up the size of the task graph that is needed for the specification of the scheduling problem, and, in an even worse way, the number of possible schedules. In [15], a subclass of UML activity diagrams is presented which is generic for the number of repetitions, and therefore suitable for the compact specification of task graphs for these manufacturing systems. The approach to reduce the complexity of the scheduling problem exploits the repetitive patterns. It reduces the original problem to a problem containing the minimum amount of identical repetitions, and after scheduling of this much smaller problem the schedule is expanded to the original size. The technique was demonstrated on a real-life example from the semiconductor industry.

Another important class of scheduling problems is that of repetitive schedules that have to satisfy hard constraints. A modelling and optimization framework for such problems is discussed in [7]. To cover a wide class of optimality criteria, an extension of the (priced) timed automata model is introduced that includes both costs and rewards as separate modelling features. A precise definition is then given of what constitutes optimal infinite behaviours for this class of models. It is shown that the derivation of optimal non-terminating schedules for such double-priced timed automata is computable. This is done by a reduction of the problem to the determination of optimal mean-cycles in finite graphs with weighted edges. Another extension is the assumption of an opponent in a game-theoretic setting (representing e.g. unmodelled effects) and the formulation of minimal cost reachability problems in the presence of such an opponent [8]. It could be proven that the optimal cost for winning such a game is computable under conditions concerning the non-zenoness of cost. Under stronger conditions (strictness of constraints) it is decidable whether there is an optimal strategy in which case an optimal strategy can be computed. These results extend previous decidability result which requires the underlying game automata to be acyclic. The results were encoded in a first prototype in HyTech which was applied on a small case-study.

In [9] the situation with partial observability is considered.

4 Interactive Modelling

In [20, 16] an alternative approach to modelling of scheduling problems is described. It is based on LSCs, a formal graphical language originally proposed for software specifications. It is shown that LSCs provide natural modeling language for systems such as the smart-card personalization machine (Case Study 3). LSCs are an example of a scenario based specification language which enables an intuitive modeling procedure using the Play-In/Play-Out approach. It is demonstrated that it is possible to synthesize a scheduler with Smart Play-Out. A model of the system in the SMV modeling language is generated which can be used to produce valid schedules or to verify schedules.

5 Evaluation, Extensions and Challenges

It was demonstrated for a large number of examples that (priced) timed automata provide an intuitive and powerful modelling framework for a broad range of scheduling problems. Significant progress towards the consideration of uncertainty and causal information structures has been made. In order to support the wider application of the TA modelling paradigm, tools for the automatic transformation of familiar descriptions of scheduling problems into TAs are under development. Modelling by TAs is much more intuitive than the formulation of equation-based models as required for MILP solvers. This translates directly into the cost of the computation of a schedule, especially if the effort for model debugging and adaptation is taken into account. As a MILP formulation may on the other hand be attractive because of the solver performance and the possibly more general formulation of cost functions, the combination of the two approaches was considered in [18]. Experience showed however that a general method for the translation of TAs into MILPs is not effective in terms of solver performance, so preference will in the future be given to tailored approaches for specific problems as e.g. job shop scheduling.

The consideration of uncertainty and risk will have to be developed further in order to get closer to solutions which are practically meaningful.

An open issue in the TA approach is to investigate which cost functions are compatible with the solution algorithms, e.g. how a combination of costs for earliness and for lateness is possible or whether the satisfaction of demands by the splitting of deliveries with a penalty for late delivery can be modelled.

While the efficiency of the approach has now been demonstrated for several medium sized case studies, the application to problems of industrial size remains a real challenge. Axxom provided a set of typical challenging scheduling problems in the process industry:

- Packaging problems were change-over costs are a dominant factor. The usual problem size is 1000 jobs on 5-10 machines and several required resources for each job.

- Synthesis of pharmaceutically relevant substances with complex production routes, coupled productions, many processing steps, need for re-work and limited storage times. Campaign formation is a major issue here. Due to long processing and resident times, 2400 jobs consisting of up to 30 tasks with 2000-5000 intermediate products on 50-200 resources may have to be considered.

- Chemical production problems where capacities of tanks to store intermediates are a major restriction; production rates must be adapted and continuous and batch productions must be synchronized. Avoidance of shut-downs of units is a major issue (similar to the problem tackled in [19]).

References

[1] Y. Abdeddaim, E. Asarin, and O. Maler, *Scheduling with timed automata*, Theoretical Computer Science (to appear), 2004.

[2] Y. Abdeddaim and O. Maler, *Preemptive job-shop scheduling using stopwatchautomata*, TACAS (J.-P. Katoen and P. Stevens, eds.), 2002.

[3] Y. Abdeddaïm, E. Asarin, and O. Maler, *On optimal scheduling under uncertainty*, Proc. TACAS (H. Gargamel and J. Hatcliff, eds.), LNCS, vol. 2619, Springer.

[4] ______, *Scheduling with timed automata*, Theoretical Computer Science, special issue selected papers from TACAS'03 (2004).

[5] Y. Abdeddaïm, A. Kerbaa, and O. Maler, *Task graph scheduling using timed automata*, FMPPTA, 2003.

[6] H.C. Bohnenkamp, H. Hermanns, R. Klaren, A. Mader, and Y.S. Usenko, *Synthesis and stochastic assessment of schedules for lacquer production*, Submitted to QEST'04.

[7] P. Bouyer, E. Brinksma, and K. G. Larsen, *Staying alive as cheaply as possible*, Proc. of 7th Int. Workshop on Hybrid Systems: Computation and Control (HSCC'2004), Lecture Notes in Computer Science, vol. 2993, Springer-Verlag, 2004, pp. 203–218.

[8] P. Bouyer, F. Cassez, E. Fleury, and K. G. Larsen, *Optimal strategies in priced timed game automata*, BRICS Report Series RS-04-4, Basic Research In Computer Science, 2004.

[9] P. Bouyer, D. D'Souza, P. Madhusudan, and A. Petit, *Timed control with partial observability*, Proc. of 15th Int. Conf. Computer Aided Verification (CAV'2003), Lecture Notes in Computer Science, vol. 2725, Springer-Verlag, 2003, pp. 180–192.

[10] M. Bozga, A. Kerbaa, and O. Maler, *Optimal scheduling of acyclic branching programs on parallel machines*, 2004, Submitted for publication.

[11] H. Dierks, G. Behrmann, and K.G. Larsen, *Solving planning problems using real-time model checking (translating pddl3 into timed automata)*, 2003.

[12] S. Engell, A. Maerkert, G. Sand, and R. Schultz, *Aggregated scheduling of a multiproduct batch plant by two-stage stochastic integer programming*, Optimization and Engineering (2004), (accepted).

[13] S. Engell and G. Sand, *A two-stage stochastic integer programming approach to real-time scheduling*, 4th Int. Conf. on Foundations of Computer-Aided Process Operations (Austin) (I. E. Grossmann and C. M. McDonald, eds.), 2003, pp. 347–350.

[14] Sebastian Engell and Sebastian Panek, *Mathematical model formulation for the axxom case study*, Tech. report, University of Dortmund, May 2003.

[15] M. Hendriks, N.J.M. van den Nieuwelaar, and F.W. Vaandrager, *Recognizing finite repetitive scheduling patterns in manufacturing systems*, Proceedings of the 1st Multidisciplinary International Conference on Scheduling: Theory and Applications (MISTA 2003), *Nottingham, UK*, Volume I (G. Kendall, E. Burke, and S. Petrovic, eds.), The University of Nottingham, 2003, ISBN 0-9545821-0-1, pp. 291–319.

[16] Hillel Kugler and Gera Weiss, *Planning a production line with LSCs*, Research report, Weizmann, 2004.

[17] A. Mader, *Towards modelling a value chain management example with uppaal - ametist case study 4*, 2003.

[18] S. Panek, O. Stursberg, and S. Engell, *Optimization of timed automata models using mixed-integer programming*, Formal Modeling And Analysis of Timed Systems, LNCS, Springer, 2004, (to appear).

[19] G. Sand and S. Engell, *Modelling and solving real-time scheduling problems by stochastic integer programming*, Computers and Chemical Engineering (2004), (to appear).

[20] Gera Weiss, *Modeling smart-card personalization machine with LSCs*, Research report, Weizmann, 2003.