Analysis and Tools: Scheduling and Planning Algorithms

VERIMAG

May 31, 2005

AMETIST DELIVERABLE 2.2.2

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/

Introduction

Scheduling and planning are not considered to be typical application domains of formal verification and fall more into the realm of disciplines such as AI and operations research as well as to specific application domains. The more general term, planning, is viewed within the AI literature as the process of finding a sequence of actions that brings a system from a given state to a desired goal state. In the AI tradition, unfortunately, the dynamics of the system is defined in a more implicit fashion using logical methods that sometime does not give a clear view of the dynamics of the system, as does automaton-based methods. A state of the system is viewed as a collection of "facts" in a data base that describe the current state of affairs. Actions are characterized by their preconditions and by their side effects, that is, the facts that they add to or remove from from the data base. This is the classical STRIPS approach whose origin is in solving puzzles in the "blocks world", a virtual world with facts like "the red block is on the blue block" and actions that put blocks on each other or remove top blocks. The AI planning community is centered around the annual conference ICAPS and holds planning competitions on problems coming from different domains. Space application are a popular source for planning problems as the distance between the earthly operator and the satelites, telescopes, spaceships and robots to be controlled is sometimes too large for tele-operations and autonomous solutions are sought.

In the OR tradition, planning refers more to some kind of "strategic" and "tactical" decision making, for example, choosing a machine replacement policy that optimizes amortized cost or an inventory handling policy. Scheduling can be viewed as a special case of planning where the action repertoire is more restricted and the decisions of the planner are about allocating bounded resources to tasks that may compete for them. The OR planning and scheduling problems tend to be more *uniform* in structure and hence can be described and solved more naturally using traditional tools of applied mathematics.[1]

When we restrict ourselves to a finite horizon, various planning and scheduling problem can be transformed into *constrained optimization* problems over a finite number of decision variables. The simpler planning and scheduling problems coming from OR translate into a more stylized format of constraints that can be solved sometimes even using simple linear programming. However most problems have a discrete decision component (for example priority between two jobs on a machine in job-shop scheduling) that renders the solution space highly non convex. Such problems are solved sometime using methods such as MILP which are more oriented toward numerical rather than logical problems. More complex planning and scheduling problems, including those treated by the AI community, have constraints with a more dominant logical part and are typically solved using constraint propagation techniques originating from constraint logic programming, or more recently by enriched SAT solvers. The alternative approach, which is the main approach pursued within the AMETIST project, consists of searching the space of paths in the dynamic transition graph implied by the system description.

Both approaches need to be upgraded in order to treat situations where the future outcome does not depend exclusively on the actions of the planner/scheduler but also on some external uncontrolled actions (the "environment" in the verification jargon. "disturbances" in the control sense). In the AI planning literature this is called *reactive planning*, and has been, in fact, subject to intense research in the context of game-playing programs. Problem with uncertainty ("adversary") are solved using algorithms that search the game graph to find a "winning" strategy. Since typically the space of possible executions grows exponentionally with the depth of the game tree, clever heuristic methods are needed to solve large problems. In the case of chess, special parallel hardware was needed to beat the human world champion. On the OR side, where problem are reduced to static optimization, it is not always easy to deal with alternating min-max non-determinsm and the uncertainty is often associated with probabilities, sometimes for historical reasons.

This report summarizes the progress in understanding, development and implementation of planning and scheduling algorithms, as well as other efforts to make connections with other disciplines and communities that occupy themselves with these issues. To avoid redundency we will *not* elaborate here much on the work reported under the case studies as well as most of the work reported in the 2nd-year deliverable 2.2.1 "control synthesis algorithms" because within the AMETIST approach controller synthesis and planning/scheduling are almost synonyms.

[1] In fact, it is a chicken and egg problem, as the availability of methematical tools often influences our choice of models and problems.

The highlights of the AMETIST work on planning and scheduling algorithms can be grouped into the following sub topics:

- Priced timed automata
- Progress in search techniques
- Timed automata and constraints
- Bridge building activities
- Other work

Priced Timed Automata

One of the major achievements of AMETIST is the development of the theoretical results, algorithms and a tool for reasoning about *priced timed automata* [13, 9, 33, 15, 16, 14, 27, 10, 11]. Before AMETIST the only cost that could be associated with a run of the automaton was the elapsed time. Priced timed automata are timed automata augmented with a cost variable that may vary at different rates at different states as well as be updated by a transition. Decidability follows from the fact that, albeit being a hybrid automaton, transition guards are not influenced by the cost variable and a finite quotient property still holds (although a more complex than for plain timed automata). During the project lifetime the theoretical results were extended to include adversaries, to treat multiple costs. The algorithmics has been improved significantly and integrated into the UPPAAL-CORA tool. It has been tested on several examples, including planning-like puzzles.

Progress in Search Techniques for Timed Automata

The approach pusrsued in [4, 6, 1, 3, 2] attempts to avoid as much as possible computation with zones by restricting the search to a finite subset of automaton ("non lazy") runs. This approach is guaranteed to find the optimum in certain deterministic problems (job-shop and task-graph included) and led to an efficient implementation. However for the case where the uncertainty is dense, for example, a task duration bounded within an interval, the simplest way to "cover" all choices of the adversary was to use a backward dynamic programming algorithm on zones, as has been done in [1, 2]. An alternative forward algorithm has been proposed in [8] based on a minimized finite-state abstraction of the automaton. Recently this idea has been improved in [20] to allow some of the minimization to be done on-the-fly, without necessarily generating the whole state space.

In [17] we tackled the problem of optimal scheduling under *discrete* uncertainty. Tasks are related by precedence constraints as well as by *conditional dependencies*, where the outcome of one task may determine whether another task should be executed. This is a natural model for scheduling programs with *if-then-else* branches on parallel machines, or for modeling faults. The whole situation is modeled as a timed game automaton where the adversary chooses the task outcomes, that is, values for certain Boolean variable that may appear in the activation condition of further tasks. The problem is then reduced to a shortest path problem in discrete weighted *game graphs*. In this work we have explored some variants of heuristic depth-first min-max search with non-chronological backtracking, instead of the breadth-first variant of heuristic search used previously, and the results are rather good: optimal solutions for problems with 20 tasks and few conditions, and sub-optimal (but good) solutions for problems with hundreds of tasks and up to 20 conditions can be found. Encouraged by the performance of depth-first search variants we tried to apply it to the problem of dense uncertainty treated in [1, 2]. We are exploring an approach based on discretizing the duration uncertainty intervals into finitely many possible durations for each task. This approach facilitates forward heuristic search but it suffers from two drawbacks: 1) The estimation of the cost of a strategy is not based on all possible adversary choices; 2) Likewise, since not all these choices were taken into account, the system may reach states for which a strategy has not been computed at all. However using the notion of an approximate strategy which takes the same decision as in the nearest point

where a strategy is defined, we can produce sub-optimal strategies, with a significant increase in performance compared to zone-based methods. Preliminary experiments of this idea have been conducted by Y. Abdeddaim and will be reported in the review meeting. Another approach for restricting the space of searched trajectories using partial-order methods has been reported in [5].

Timed Automata and Constraints

Although optimization techniques differ from timed automata reachability algorithms, they share the type of constraints needed for computation, namely difference constraints of the form $x - y < c$ that specify bounds on distance between events. Although the main thrust of the project was to suggest TA techniques as an *alternative* to the constrained optimization approach, we have also explored ways to combine both. In a series of works [31, 30, 32] techniques based on MILP have been integrated into the search mechanism for timed automata and have been applied to scheduling problems. A new SAT solver for difference constraints has been developed in [21] as replacement for the previous solver developed also within AMETIST [28]. The performance of the last version of the solver [**?**] is much improved and it can find the optimal solutions of some notorious job-shop scheduling problems such as ABZ5 and FT10.

Bridge-building Activities

Members of AMETIST made efforts to push the timed automaton vision to relevant communities. O. Maler and Y. Abdeddaim have presented the results of [6, 1, 3] in affiliated workshops of ICAPS'02 in Toulouse, and a year later O. Maler gave a one-day tutorial in ICAPS'03 in Trento about timed automata and scheduling. This year K. Larsen will give an invited talk at ICAPS'05 on real-time planning using timed automata. The work on the NASA K9 Rover case-study [12] has demonstrated the applicability of timed automata for space application, and led to a discussion with labs such as LAAS, Toulouse about closer collaboration between the AI planning crowd and the timed automaton community. As an opening toward the OR community, preparations for a new Dagstuhl seminar on different approaches to scheduling under uncertainty, by R. Mohring and O. Maler have started. The unifying model for controller synthesis, planning and scheduling [29] has been presented to a variety of communities such as those of discrete-event systems (WODES) and control (CC project).

Other Work

Other scheduling-related work, already mentioned in previous deliverables, includes [7, 36, 25, 24, 19, 18] on specializing timed automata for scheduling of embedded real-time software, and the work on real-time stochastic scheduling in [34, 23, 22, 35], as well as work on applying LSCs [26].

References

[1] Y. Abdeddaïm, E. Asarin, and O. Maler. On optimal scheduling under uncertainty. In H. Gargamel and J. Hatcliff, editors, *Proc. TACAS*, volume 2619 of *LNCS*. Springer, 2003.

[2] Y. Abdeddaïm, E. Asarin, and O. Maler. Scheduling with timed automata. *Theoretical Computer Science*, 2005. to appear.

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

[4] Y. Abdeddaïm and O. Maler. Preemptive job-shop scheduling using stopwatchautomata. In J.-P. Katoen and P. Stevens, editors, *TACAS*, 2002.

[5] Y. Abdeddaïm and P. Niebert. On the use of partial order methods in scheduling. In *Ninth International Conference on Project Management and Scheduling (PMS 04)*, 2004. to be published as online abstract.

[6] Yasmina Abdeddaïm. *Scheduling with Timed Automata*. PhD thesis, INPG Grenoble, November 2002.

[7] K. Altisen, G. Goessler, and J. Sifakis. Scheduler modeling based on the controller synthesis paradigm. *Journal of Real-Time Systems, special issue on Control Approaches to Real-Time Computing*, 23:55–84, 2002.

[8] K. Altisen and S. Tripakis. Tools for controller synthesis of timed systems. In *RT-TOOLS*, 2002.

[9] G. Behrmann. Guiding and cost optimizing uppaal. Web-page, 2002.

[10] G. Behrmann, K.G. Larsen, and J.I. Rasmussen. Optimal scheduling using priced timed automata. *Performance Evaluation Review, ACM Sigmetric*, 2005. To appear.

[11] G. Behrmann, K.G. Larsen, and J.I. Rasmussen. Priced timed automata: Algorithms and applications. In *Proceedings of FMCO'04*, Lecture Notes in Computer Science. Springer Verlag, 2005. To appear.

[12] S. Bensalem, M. Bozga, M. Krichen, and S. Tripakis. Testing conformance of real-time applications by automatic generation of observers. In *Runtime Verification (RV'04)*, 2004.

[13] 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.

[14] Patricia Bouyer, Ed Brinksma, and Kim G. Larsen. Staying alive as cheaply as possible. In Rajeev Alur and George J. Pappas, editors, *Proceedings of the 7th International Conference on Hybrid Systems: Computation and Control (HSCC'04)*, volume 2993 of *Lecture Notes in Computer Science*, pages 203–218, Philadelphia, Pennsylvania, USA, March 2004. Springer-Verlag.

[15] Patricia Bouyer, Franck Cassez, Emmanuel Fleury, and Kim G. Larsen. Optimal strategies in priced timed game automata. In Kamal Lodaya and Meena Mahajan, editors, *Proceedings of the 24th Conference on Fundations of Software Technology and Theoretical Computer Science (FSTTCS'04)*, volume 3328 of *Lecture Notes in Computer Science*, pages 148–160, Chennai, India, December 2004. Springer.

[16] Patricia Bouyer, Franck Cassez, Emmanuel Fleury, and Kim G. Larsen. Synthesis of optimal strategies using HyTech. In Luca De Alfaro, editor, *Proceedings of the Workshop on Games in Design and Verification (GDV'04)*, volume 119 of *Electronic Notes in Theoretical Computer Science*, pages 11–31, Boston, Massachusetts, USA, February 2005. Elsevier Science Publishers.

[17] M. Bozga, A. Kerbaa, and O. Maler. Optimal scheduling of acyclic branching programs on parallel machines. In *Real-Time Systems Symposium (RTSS)*, pages 208–217. IEEE Press, 2004.

[18] P. Caspi, A. Curic, A. Maignan, C. Sofronis, and S. Tripakis. Translating discrete-time Simulink to Lustre. In *Embedded Software (EMSOFT'03)*, volume 2855 of *LNCS*. Springer, 2003.

[19] P. Caspi, A. Curic, A. Maignan, C. Sofronis, S. Tripakis, and P. Niebert. From Simulink to SCADE/Lustre to TTA: a layered approach for distributed embedded applications. In *Languages, Compilers, and Tools for Embedded Systems (LCTES'03)*. ACM, 2003.

[20] F. Cassez, A. David, E. Fleury, K.G. Larsen, and D. Lime. Efficient on-the-fly algorithms for the analysis of timed games. In Luca de Alfaro Martin Abadi, editor, *To appear in Proceedings of CONCUR 2005*, Lecture Notes in Computer Science. Springer Verlag, 2005.

[21] S. Cotton, E. Asarin, O. Maler, and P. Niebert. Some progress in satisfiabilty checking for difference logic. In *FORMATS/FTRTFT'04*, number 3253 in LNCS, pages 263–276. Springer, 2004.

[22] 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*, 5:335–359, 2004.

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

[24] Ch. Kloukinas and S. Yovine. Synthesis of safe, qos extendible, application specific schedulers for heterogeneous real-time systems. In *Proceedings of '5th Euromicro Conference on Real-Time Systems (ECRTS'03)'*, Porto, Portugal, July 2003.

[25] Christos Kloukinas, Chaker Nakhli, and Sergio Yovine. A methodology and tool support for generating scheduled native code for real-time Java applications. In Rajeev Alur and Insup Lee, editors, *EMSOFT 2003*, volume 2855 of *Lecture Notes in Computer Science*, pages 274–289, Philadelphia, Pennsylvania, USA, October 2003. Springer-Verlag.

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

[27] Kim G. Larsen and Jacob I. Rasmussen. Optimal conditional reachability for multi-priced timed automata. In *In Proceedings of FoSSACS 2005*, number 3441 in Lecture Notes in Computer Science, pages 234–249, 2005.

[28] Moez Mahfoudh. *On Satisfaiblity Checking for Difference Logic*. PhD thesis, UJF Grenoble, May 2003.

[29] O. Maler. On optimal and sub-optimal control in the presence of adversaries. In *Workshop on Discrete Event Systems (WODES)*, pages 1–12. IFAC, 2004. Invited talk.

[30] S. Panek, O. Stursberg, and S. Engell. Job-shop scheduling by combining reachability analysis with linear programming. In *Proc. 7th Int. Workshop on Discrete Event Systems*, pages 199–204, 2004.

[31] S. Panek, O. Stursberg, and S. Engell. Optimization of timed automata models using mixed-integer programming. In *Formal Modeling And Analysis of Timed Systems*, volume 2791 of *LNCS*, pages 73–87. Springer, 2004.

[32] S. Panek, O. Stursberg, and S. Engell. Efficient synthesis of production schedules by optimization of timed automata. *Control Engineering Practice*, 2005. submitted.

[33] J. Rasmussen, K. G. Larsen, and K. Subramani. Resource-optimal scheduling using priced timed automata. In *Proc. 10th Int. Conf. of Tools and Algorithms for the Construction and Analysis of Systems (TACAS'2004)*, volume 2988 of *Lecture Notes in Computer Science*, pages 220–235. Springer-Verlag, 2004.

[34] G. Sand and S. Engell. Aggregated batch scheduling in a feedback structure. In J. v. Schijndel and J. Grievink, editors, *European Symp. on Computer Aided Process Engineering-12*, volume 10 of *Computer-Aided Chemical Engineering*, pages 775–780. Elsevier Science, 2002.

[35] G. Sand and S. Engell. Risk conscious scheduling of batch processes. In *Proc. Computer-Aided Chemical Engineering*, pages 588–593, 2004.

[36] J. Sifakis, S. Tripakis, and S. Yovine. Building models of real-time systems from application software. In *Proceedings of the IEEE Special issue on modeling and design of embedded*, pages 91(1):100–111, January 2003.