Deliverable D2.1

Efficient algorithms for multi-player games with quantitative aspects

Gilles Geeraerts (ULB), Samuel Dehouck (ULB), Kim Larsen (AAU), Nicolas Markey (CNRS)

Project	Cassting — Collective Adaptive Systems SynThesIs with Non-zero-sum Games		
Project id.	FP7-601148		
Workpackage	WP2	Nature	R (report)
Deliverable	D2.1	Dissemination	PU (public)
Date submitted	March 2015	Date due	March 2015
Version	1.0		

;

Efficient algorithms for multi-player games with quantitative aspects

Gilles Geeraerts (ULB), Samuel Dehouck (ULB),
Kim Larsen (AAU), Nicolas Markey (CNRS)

March 31, 2015

1 Introduction and scientific background

Rich game models for CAS Modeling and reasoning on the behaviours of Collective Adaptive Systems (CAS) is a notoriously difficult task. Two important reasons that make the behaviours of CAS difficult to grasp are:

1. the fact that those behaviours very often depend on *measurable parameters*, i.e. quantities, and not purely Boolean values. Many CAS exhibit behaviours that are highly dependent on the timing of the actions, hence *time* is a very important quantity one has to deal with when analysing those systems. Another one is *energy*, because many CAS are actually autonomous embedded systems, that rely on limited energy sources such as batteries. Besides, randomised aspects may also be an important characteristics of our models.

2. the fact that a CAS is made up of many agents, whose interaction is often complex. In this case, agents do not always compete strictly against each other, but all try to maximise their own profit.

Thus, in order to develop a comprehensive game-theoretic theory allowing to reason about CAS, we need to reason on models and techniques that capture precisely those characteristics. In previous contributions of the project (see in particular Deliverable 1.1), we have proposed such new models. Let us recall briefly those models:

1. A very simple quantitative model for computer systems in general and CAS in particular is that of *weighted graphs*. In this model, each node of the graph represents a potential state of the system, and each edge a possible action, with an associated *cost*. This cost can model energy consumption, for instance, and can be, in general, any rational value. Then, one can adapt the classical notion of *game played on graph* to such weighted graphs. In such games, one is usually interested in computing *optimal strategies* for both players, i.e., strategies that guarantee optimal costs of the plays whatever the other players are doing.

2. Arguably one of the most important quantities for CAS is *time*. A classical model to express explicitly the elapsing of time is that of *timed automata*,

i.e., finite automata augmented with a finite set of real-values clocks that can be tested and reset, and for which weighted extensions have been proposed [ALP01, BFH+01, BBBR07]. By mixing game and weighted extensions of timed automata, the class of *priced timed games* is naturally obtained, where two players compete over a weighted timed automaton. In the turn-based case, locations are partitioned into both players. During a play, a token is moved in the timed automaton. When the token is in a location of a certain player, he has to react by choosing a delay and a transition that will be fired after letting the time elapse. The play goes until the target set of states is hopefully reached: in that case, player 1 wins with a cost given by the cost of the run in the weighted timed automaton. Otherwise, player 1 loses. Player 1 wants to win with a minimum cost, whereas player 2 wants to make player 1 lose or win with the greatest cost possible. These games have first been studied and solved in a restricted case where all cycles in the configuration graph of the weighted timed automaton have a weight bounded below by a fixed positive constant [ABM04, BCFL04]. This restriction, called *strictly non-Zeno costs*, ensures that the optimal value that player 1 can ensure against all strategies of player 2 is computable in exponential time, both with an optimal strategy for player 1.

Unfortunately, dropping this constraints on cycles of the automaton leads to undecidability results. For instance, it has been shown in [BBR05, BBM06] that the problem of knowing whether player 1 has a strategy ensuring a given cost cannot be decided in general. It is already true for weighted timed games with three clocks and only non negative weights over locations and transitions.

Further work has been done in order to recover decidability results in the case of games without the strictly non-Zeno costs restriction. In particular, an algorithm is known that computes the optimal cost for player 1 in turn-based weighted timed games with one clock and only non-negative costs over locations: a first algorithm, given in [BLMR06a], runs in 3-EXPTIME. This complexity has then be lowered to a single exponential by refining the algorithm in [Rut11], and a different approach, closer to policy iteration algorithms for Markov decision processes, has also been proposed by [HIJM13] leading to the same complexity. The actual complexity of this problem is currently not known. It has to be noticed that optimal strategies may not exist for player 1 in these games. However, those algorithms also permit to compute *almost-optimal* strategies, i.e., for all arbitrary positive ε, the algorithm may compute a strategy for player 1 ensuring him to win with a cost not greater than the optimal cost translated with ε.

3. Of prime importance for embedded computer systems and for CAS are the several probabilistic (quantitative) extensions of traditional models. On of the most classical model in this setting is that of *Markov Decision Process* (MDP), which are finite non-deterministic automata, where the transition function δ associates, to each pair (state, action), a *probability distribution* over the possible successor states. The semantics of MDP can be interpreted in terms of a game between one player who chooses the next action a to play from the current state s, and where the second player chooses the successor state according to the probability distribution $\delta(s, a)$.

A classical problem on MDPs is to synthesise a *strategy*, i.e. a function indicating the first player which action to play after each finite play prefix, in order to ensure some probabilistic objective (for instance, reaching a designated goal state with high probability).

Efficient algorithms for synthesising CAS Because those models are *very rich*, their related analysis problems usually have very high complexity. Hence the need for *efficient algorithms and tools* to analyse those models, that rely on *dedicated data structures*. The aim of this deliverable is to present the latest contributions of the CASSTING project wrt to these topics.

Content The contributions presented in this deliverable can be summarised as follows:

Timed systems We present efficient techniques to: (1) compute optimal and ε-optimal winning strategies in priced timed games (contributions 1 and 2), with some implementation work. (2) perform efficient synthesis of controllers where the objectives are specified using a fragment the MTIL real-time logic (contribution 3) (3) perform robust synthesis of controller in timed systems (contributions 4 and 5) and (4) synthesise runs in timed automata and weighted timed automata under particular energy or synchronisation constraints (contributions 6 and 7). Finally, we also introduce novel efficient data structures to perform verification and synthesis of timed systems (contribution 8).

Probabilistic systems We present efficient techniques and tools to synthesise optimal strategies in MDPS (contribution 9). We also introduce the notion of *beyond worst-case synthesis* for MDPs, which allows one to obtain optimal strategies meeting both worst-case and average case objectives, which makes a lot of sense in practice for CAS. We provide efficient techniques to solve those problems (contribution 10).

Other quantitative models We show how to compute secure equilibria in weighted games, and show that we can synthesise efficient strategies (i.e., finite memory strategies with few memory) to enforce such equilibria (contribution 11).

Finally, we close this report by sketching ongoing research on energy games, priced timed games and synthesis of MITL.

2 Contributions – Timed models

Contribution 1: Computing optimal winning strategies in one-clock priced timed games

Timed automata are a powerful formalism for modelling real-time systems, i.e., systems whose specifications heavily depend on timing information. Timed automata were introduced in the late 1980's [AD90], and have been extensively studied since then: they enjoy both rich expressiveness and relatively efficient algorithms, which have been implemented in highly optimized tools like Uppaal.

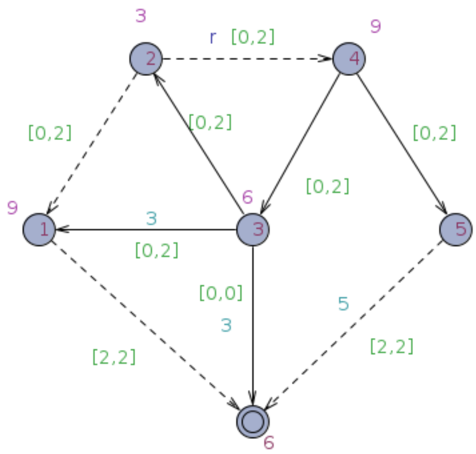

Figure 1: An example of priced timed game, shown in the GUI of UppAal, and ready to be analysed by our tool. The r on the edge from location 2 to location 4 indicates that the clock is reset on this edge. The natural numbers next to the locations are the rates; the weights on the edges are the discrete weights and the intervals on the edges are their respective guards. Controllable edges are solid, uncontrollable edges are dashed.

In order to model more quantities besides time, timed automata have been extended as *priced* (or *weighted*) *timed automata*. In those models, the extra variables are just *observers*: they can measure certain (nondecreasing) quantities, but the behaviour of the model only depends on clock values. Again, those models enjoy very nice algorithmic properties, and optimal reachability, for instance, is decidable.

When further extending priced timed automata to games, the decidability frontier is reached. Actually, it was shown that deciding the winner of a priced timed games is undecidable for models with three clocks. On the other hand, the problem was proved decidable when restricted to one clock [BLMR06b], but with high complexity. Recently, the algorithm was improved [HIM13] to exponential-time worst case complexity, which is reasonable to implement.

We have implemented this algorithm and run a set of benchmarks to try and better understand its actual running time. The idea behind the algorithm is that solving a priced timed game can be reduced to solving several so-called simple priced timed games in which there is no reset and in which all constraints on the clock are of the same type. It has been proved that this reduction can be done in polynomial time.

Then, to solve a simple priced timed game, the algorithm reduces it to solving several priced games (i.e., *untimed* games). The number of such priced games one needs to solve is at most exponential in the number of states of the original simple priced timed game.

Finally, solving those priced games can be done in polynomial time thanks to, either an extension of Dijkstra's classical algorithm for shortest path in a graph, or a strategy iteration technique.

Our tool takes as input a priced timed game generated by Uppaal (using the

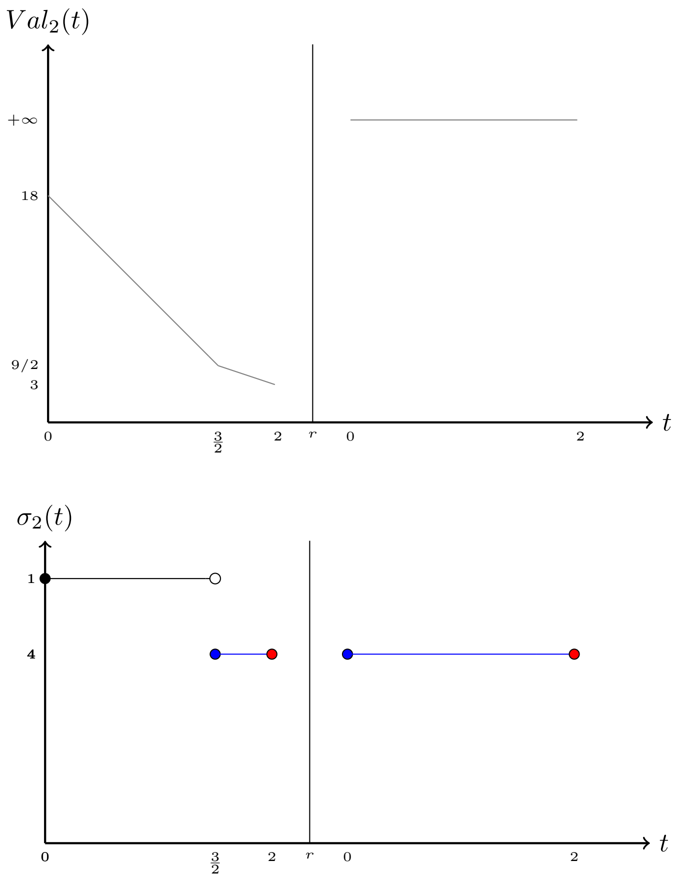

Figure 2: An example output of the tool when applied to the priced timed game in Figure 1. Since the game contains a reset, solving it reduces to solving two priced timed games without reset (not shown). The optimal value functions achieved from location 2 in those games are shown respectively on the left an right of the top figure. Bottom figure shows the associated strategies. From this information, our tool rebuilds the optimal value function.

XML format used by Uppaal, thus benefiting from the Upaal GUI), and outputs cost functions in a PDF file. Figure 1 shows an example of priced timed game and Figure 2 shows an excerpt of the output of our tool on this example.

In terms of efficiency, we were not able to create examples for which the run-time would grow exponentially. But in spite of our efforts, we were not able (yet) to prove that the algorithm actually runs in polynomial time. This is our current conjecture, and we are trying to prove it formally.

- Implementation available at: `https://github.com/samueldehouck/PTGs`
- Partners: ULB, CNRS.

Contribution 2: Computing ϵ-optimal winning strategies in priced timed games

While for the general case (unrestricted number of clocks), the problem of computing the optimal price with which a target location can be reached is undecidable, we recently prove that (under some restrictions of the structure of the automaton) this value can be approximated.

Such a result was already known for the special case of *strongly non-Zeno* priced timed automata: in such automata, the cost variable is required to increase by at least 1 unit along any cycle (of the region automaton). The decidability of this case is obvious, as it suffices to unwind the region automaton to an appropriate depth, and to check the property in the resulting finite tree structure, in order to get the result.

Our approximation result works on a larger class of priced timed games, where the requirement is that along any cycle of the region automaton, either the cost is constant, or it is increased by at least 1 unit. The technique described above obviously fails, and we proved that actually the optimal cost cannot be computed. In order to approximate the optimal cost, our algorithm relies on a partial unfolding of the region automaton, where the connected components where the cost is constant are not unfolded. In those components, an approximate value of the optimal cost can be computed for each configuration (in a backward manner, i.e., assuming that we can compute the optimal cost at the nodes leaving such components). We perform an exact computation along the branches where the cost strictly increases: we can bound the length of such branches thanks to our original assumption. In the end, we get a doubly-exponential approximation algorithm.

These results were obtained by CNRS. They can be found in a technical report [BJM14], and are currently under submission.

- Technical report available at: Patricia Bouyer, Samy Jaziri, and Nicolas Markey. **On the value problem in weighted timed games**. Research Report LSV-14-12, Laboratoire Spécification et Vérification, ENS Cachan, France, October 2014. 24 pages.
- Partners: CNRS.

Contribution 3: Efficient synthesis for a fragment of MTL

In this paper we offer an efficient controller synthesis algorithm for assume-guarantee specifications of the form $\phi_1 \wedge \phi_2 \wedge \cdots \wedge \phi_n \to \psi_1 \wedge \psi_2 \wedge \cdots \wedge \psi_m$. Here, ϕ_i, ψ_j are all safety-$\text{MTL}_{0,+\infty}$ properties, where the sub-formulas ϕ_i are supposed to specify assumptions of the environment and the sub-formulas ψ_j are specifying requirements to be guaranteed by the controller. Our synthesis method exploits the engine of Uppaal-Tiga and the novel translation of safety- and co-safety-$\text{MTL}_{0,+\infty}$ properties into under-approximating, deterministic timed automata. Our approach avoids determinization of Búchi automata, which is the main obstacle for the practical applicability of controller synthesis for linear-time specifications. The experiments demonstrate that the chosen specification formalism is expressive enough to specify complex behaviors. The proposed approach is sound but not complete. However, it successfully produced solutions for all the experiments. Additionally we compared our tool with Acacia+ and

Unbeast, state-of-the-art LTL synthesis tools; and our tool demonstrated better timing results, when we applied both tools to the analogous specifications.

- Work published in: Peter E. Bulychev, Alexandre David, Kim G. Larsen, Guangyuan Li: **Efficient controller synthesis for a fragment of $\mathbf{MTL}_{0,+\infty}$**. Acta Inf. 51(3-4): 165-192 (2014)

- Partners: Aalborg

Contribution 4: Robust synthesis for real-time systems

Specification theories for real-time systems allow reasoning about interfaces and their implementation models, using a set of operators that includes satisfaction, refinement, logical and parallel composition. To make such theories applicable throughout the entire design process from an abstract specification to an implementation, we need to reason about the possibility to effectively implement the theoretical specifications on physical systems, despite their limited precision. In the literature, this implementation problem has been linked to the robustness problem that analyzes the consequences of introducing small perturbations into formal models.

We address this problem of robust implementations in timed specification theories. We first consider a fixed perturbation and study the robustness of timed specifications with respect to the operators of the theory. To this end we synthesize robust strategies in timed games. Finally, we consider the parametric robustness problem and propose a counter-example refinement heuristic.

- Work published in: Kim G. Larsen, Axel Legay, Louis-Marie Traonouez, Andrzej Wasowski, **Robust synthesis for real-time systems**. Theor. Comput. Sci. 515: 96-122 (2014)

- Partners: Aalborg (in a collaboration with INRIA).

Contribution 5: Robust control of sampled switched systems with partial observability

(Sampled) switched systems are a convenient formalism for modeling physical processes interacting with a digital controller. Unfortunately, the formalism does not capture the distributed nature encountered e.g. in cyber-physical systems, which are organized as networks of elements interacting with each other and with *local* controllers. Most current methods for control synthesis can only produce a centralized controller, which is assumed to have complete knowledge of all the component states and can interact with all of them.

Among these existing methods, we focused on the approach of [FKS14]: this solution is based on the decomposition by repeated bisection of the region of interest surrounding the objective point. Each resulting tile of the bisection is associated with a sequence of modes (or pattern) that maps the tile inside the zone of interest. More precisely, a sequence of modes is *valid* if the original tile is mapped into the *secure zone* for all intermediary points of the sequence, and is mapped into the *target zone* at the end of the sequence. See Fig. 3 for an illustration of the technique.

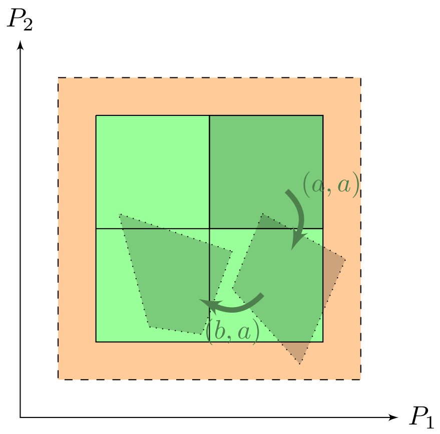

Figure 3: Example of a valid pattern from the upper right tile: playing action (a, a) (which prescribes both players to play action a) remains in the secure (green+orange) zone; then playing action (b, a) (i.e., Player 1 plays action b and Player 2 plays action a) takes the whole tile back within the target zone (green), as required. A system is controllable if all the tiles have a valid pattern; when this is not the case, the tile can be further decomposed (or the systems is declared uncontrollable if the maximal decomposition depth has been reached).

While this approach indeed synthesizes a controller, it does not involve a *game* between several components: the controller simply generates one trace of the system. This assumes that it has perfect observation and control over the whole system. This is not always very realistic, and is not very robust as it gives no guarantee in case one of the components of the system does not behave as expected.

We extended this method in order to address these drawbacks. We first considered the problem of *robustness*, trying to take into account the possible failures of some component(s). In this approach, a pattern for a single player is valid if, when combined with *any* pattern of the other players, the projection of the resulting tiles remain within the objectives of that player. Figure 4 illustrates this approach.

This approach enjoys the following properties: after playing the pattern of a given player, the projection of the resulting tile is in the projection of that player's objective. As a consequence, if all the players play valid patterns *of the same length* from a given tile, then the resulting tile is included in the target zone (and all the intermediary tiles remain in the secure zone). Additionally, if for some reason some player(s) do not follow their strategies, then the players who did follow their strategies will satisfy their objective at the end of that pattern (but the system may end up outside the target zone, where no player has any guarantee).

The controllers computed above are, in some sense, distributed, but they have to observe the states of all the components. Additionally, it is required that all patterns have the same length, as otherwise, the system may end up in the secure zone (as opposed to the target zone), where no controller has been computed.

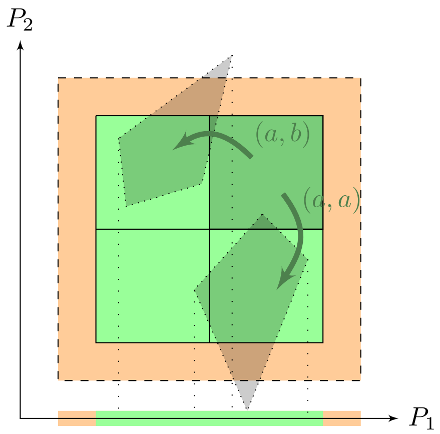

Figure 4: Example of a valid pattern for Player 1. The pattern is made of the single action a: it is valid because both patterns (a, a) and (a, b) lead to tiles whose projection is in the projection of the target zone for Player 1. Deciding the validity of longer patterns requires exploring a tree resulting of possible completions of the pattern being checked.

In order to circumvent these two restrictions, we considered *local observation*: there we assume that each player can only observe part (or possibly none) of the state of the other components. The variables that are not observed are assumed to take any value in their secure zone. The technique applied is then similar to the above: for each pattern, we explore the whole tree of combinations, until a valid pattern is obtained. Figure 5 illustrates this technique.

There are two benefits from this approach: first, the controller has only partial observation of the whole system, which is very relevant in many situations. Additionally, under suitable hypotheses, we get rid of the restriction about the patterns having the same length: a pattern is winning even if starting from the secure zone of the unobserved variables, i.e., even if played while another player is in the middle of his pattern. Then provided that all the players follow their pattern, whatever their lengths, the system is guaranteed to visit the target zone of each player infinitely often (but notice that it might never end up in the whole target zone itself).

We implemented this approach on top of the tool MINIMATOR [Küh], and applied it to a model of a two-room floor-heating system (see Fig. 6).

The heaters will start to heat up to temperature T_f when switched on. The state $X = (H_1, T_1, H_2, T_2)^T$ of this model is formed by the temperatures of the two rooms (T_1, T_2) and the heaters (H_1, H_2). The dynamics of the model can be described by the equation $\dot{X} = A_u X + B_u$ with

$$A_u = \begin{pmatrix} -\beta_1 - u_1\alpha_f & \beta_1 & 0 & 0 \\ \gamma_1 & -\alpha_{e1} - \gamma_1 - \alpha_{21} & 0 & \alpha_{21} \\ 0 & 0 & -\beta_2 - u_2\alpha_f & \beta_2 \\ 0 & \alpha_{12} & \gamma_2 & -\alpha_{e2} - \gamma_2 - \alpha_{12} \end{pmatrix}, \quad B_u = \begin{pmatrix} u_1\alpha_f T_f \\ \alpha_{e1} T_e \\ u_2\alpha_f T_f \\ \alpha_{e2} T_e \end{pmatrix},$$

where $u_1, u_2 \in \{0, 1\}$ indicate the state of the heaters $(0 = \text{off}, 1 = \text{on})$. In our model, when turned on, the heating system slowly gets warmer (up to a maximal

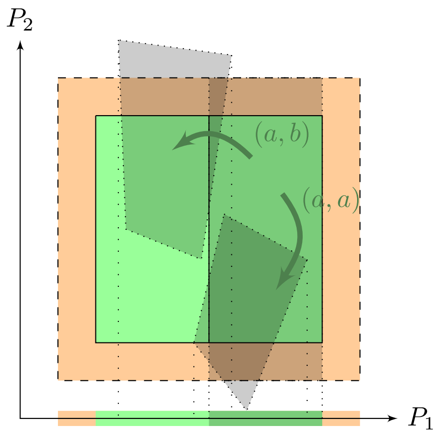

Figure 5: Example of a valid pattern for Player 1 with local observation. The pattern maps the whole stripe on the right to tiles whose projection is within the objective of Player 1.

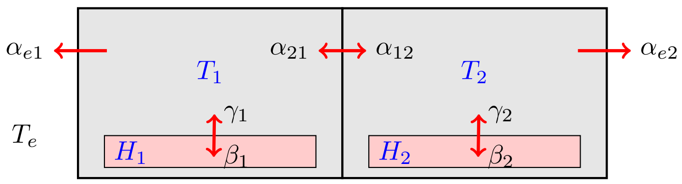

Figure 6: A two-room water underfloor heating system.

temperature of $T_f = 40°C$), and then releases heat to the room. Heat exchange also occur between both rooms, and with the outside (outside air temperature: $T_e = 10°C$). We could synthesize a controller in the following cases:

- in case both controllers have access to the temperature of the room and of the heater they control (Figure 7 displays the resulting controller for this situation);
- in case they only observe the local temperature;
- in case they observe the local and the outside temperature.

- Work published in: Laurent Fribourg, Ulrich Kühne, and Nicolas Markey. **Game-based synthesis of distributed controllers for sampled switched systems**. In Proceedings of the 2nd International Workshop on Synthesis of Continuous Parameters (SYNCOP'15), Electronic Proceedings in Theoretical Computer Science, London, UK, April 2015. To appear.
- Partners: CNRS (in a collaboration with the university of Bremen, Germany)

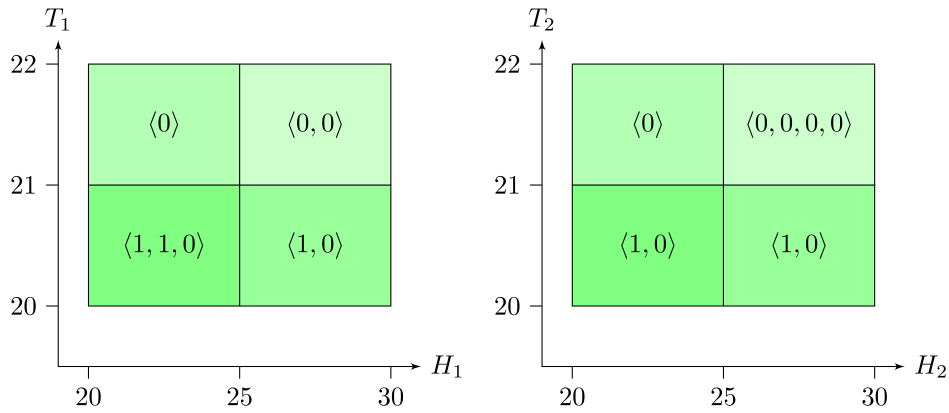

Figure 7: Controllers with partial observation (local room and heater temperature)

Contribution 6: Synchronizing Words for Weighted and Timed Automata

The problem of synchronizing automata is concerned with the existence of a word that sends all states of the automaton to one and the same state. This problem has classically been studied for complete deterministic finite automata, with the existence problem being NLOGSPACE-complete.

In this paper we consider synchronizing-word problems for weighted and timed automata. We consider the synchronization problem in several variants and combinations of these, including deterministic and non-deterministic timed and weighted automata, synchronization to unique location with possibly different clock valuations or accumulated weights, as well as synchronization with a safety condition forbidding the automaton to visit states outside a safety-set during synchronization (e.g. energy constraints). For deterministic weighted automata, the synchronization problem is proven PSPACE-complete under energy constraints, and in 3-EXPSPACE under general safety constraints. For timed automata the synchronization problems are shown to be PSPACE-complete in the deterministic case, and undecidable in the non-deterministic case

- Work published in: Laurent Doyen, Line Juhl, Kim Guldstrand Larsen, Nicolas Markey, Mahsa Shirmohammadi, **Synchronizing Words for Weighted and Timed Automata**. FSTTCS 2014: 121-132
- Partners: Aalborg, CNRS.

Contribution 7: Synthesising runs in timed automata under quantitative constraints

We investigate a number of problems related to infinite runs of weighted timed automata, subject to lower-bound constraints on the accumulated weight. Closing an open problem from an earlier paper, we show that the existence of an infinite lower-bound-constrained run is – for us somewhat unexpectedly – undecidable for weighted timed automata with four or more clocks. This undecidability result assumes a fixed and known initial credit. We show that the related problem of existence of an initial credit for which there exists a feasible run is decidable in PSPACE. We also investigate the variant of these problems where only bounded-duration runs are considered, showing that this restriction makes our original problem decidable in NEXPTIME. Finally, we prove that the universal versions of all those problems(i.e, checking that all the considered runs satisfy the lower-bound constraint) are decidable in PSPACE.

- Work published in: Patricia Bouyer, Kim G. Larsen, Nicolas Markey: **Lower-bound-constrained runs in weighted timed automata**. Perform. Eval. 73: 91-109 (2014)
- Partners: Aalborg, CNRS

Contribution 8: Memory Efficient Data Structures for Explicit Verification of Timed Systems

Timed analysis of real-time systems can be performed using continuous (symbolic) or discrete (explicit) techniques. The explicit state-space exploration can be considerably faster for models with moderately small constants, however, at the expense of high memory consumption. In the setting of timed-arc Petri nets, we explore new data structures for lowering the used memory: PTries for efficient storing of configurations and time darts for semi-symbolic description of the state-space. Both methods are implemented as a part of the tool TAPAAL and the experiments document at least one order of magnitude of memory savings while preserving comparable verification times.

- Work published in: Peter Gjøl Jensen, Kim Guldstrand Larsen, Jirí Srba, Mathias Grund Sørensen, Jakob Haahr Taankvist, **Memory Efficient Data Structures for Explicit Verification of Timed Systems**. NASA Formal Methods 2014: 307-312
- Partners: Aalborg.

3 Contributions – Probabilistic models

Contribution 9: Efficient algorithms and tools for MDPs

When treating Markov decision processes (MDPs) with large state spaces, using explicit representations quickly becomes unfeasible. Lately, Wimmer et al. have proposed a so-called symblicit algorithm for the synthesis of optimal strategies in MDPs, in the quantitative setting of expected mean-payoff. This algorithm, based on the strategy iteration algorithm of Howard and Veinott, efficiently combines symbolic and explicit data structures, and uses binary decision diagrams as symbolic representation. The aim of this paper is to show that the new data structure of pseudo-antichains (an extension of antichains) provides another interesting alternative, especially for the class of monotonic MDPs. We design efficient pseudo-antichain based symblicit algorithms (with open source implementations) for two quantitative settings: the expected mean-payoff and the stochastic shortest path. For two practical applications coming from automated planning and LTL synthesis, we report promising experimental results w.r.t. both the run time and the memory consumption.

- Work published in: Aaron Bohy, Véronique Bruyère, Jean-François Raskin, **Symblicit algorithms for optimal strategy synthesis in monotonic Markov decision processes.** SYNT 2014: 51-67
- Partners: ULB, UMONS.

Contribution 10: Beyond worst-case synthesis

Classical analysis of two-player quantitative games involves an adversary (modeling the environment of the system) which is purely antagonistic and asks for strict guarantees while Markov decision processes model systems facing a purely randomized environment: the aim is then to optimize the expected payoff, with no guarantee on individual outcomes. We introduce the beyond worst-case synthesis problem, which is to construct strategies that guarantee some quantitative requirement in the worst-case while providing an higher expected value against a particular stochastic model of the environment given as input. We consider both the mean-payoff value problem and the shortest path problem. In both cases, we show how to decide the existence of finite-memory strategies satisfying the problem and how to synthesize one if one exists. We establish algorithms and we study complexity bounds and memory requirements.

- Work published in:
  1. Véronique Bruyère, Emmanuel Filiot, Mickael Randour, Jean-François Raskin,, **Meet Your Expectations With Guarantees: Beyond Worst-Case Synthesis in Quantitative Games**. STACS 2014: 199-213
  2. Véronique Bruyère, Emmanuel Filiot, Mickael Randour, Jean-François Raskin: **Expectations or Guarantees? I Want It All! A crossroad between games and MDPs**. SR 2014: 1-8
- Partners: CNRS, ULB, UMONS.

4 Contributions – Other quantitative models

Contribution 11: Secure equilibria in weighted games

We consider two-player non zero-sum infinite duration games played on weighted graphs. We extend the notion of secure equilibrium introduced by Chatterjee et al., from the Boolean setting to this quantitative setting. As for the Boolean setting, our notion of secure equilibrium refines the classical notion of Nash equilibrium. We prove that secure equilibria always exist in a large class of weighted games which includes common measures like sup, inf, lim sup, lim inf, mean-payoff, and discounted sum. Moreover we show that one can synthesize finite-memory strategy profiles with few memory. We also prove that the constrained existence problem for secure equilibria is decidable for sup, inf, lim sup, lim inf and mean-payoff measures. Our solutions rely on new results for zero-sum quantitative games with lexicographic objectives that are interesting on their own right.

- Work published in: Véronique Bruyère, Noémie Meunier, Jean-François Raskin: **Secure equilibria in weighted games**. CSL-LICS 2014: 26
- Partners: ULB, UMONS.

5 On-going work

5.1 Average-energy games

Energy games are games played on a weighted graph, where the weights can be either positive or negative. They can be used to model various quantities, such as the energy needed or the price to pay for doing some actions.

Mean-payoff games aims at finding strategies that minimize the average price to pay along an execution. They have been extensively studied, in particular because of the exact complexity of solving them is open.

Energy games are based on the same model, but the problem there is to find a strategy for reaching a given target (or for running ad infinitum) while keeping the energy level above a given threshold. This kind of problems have been extensively studied over the last five years.

In a recent, yet unpublished work, we considered the problem mixing both constraints. Actually, we introduced a variant of the mean-payoff games where the aim is to minimize the average value of the accumulated cost along the run. Formally, given a sequence of weights $(m_i)_{i\in\mathbb{N}}$, while mean-payoff games aims to minimize the limit

$$\liminf_{n\to\infty} \frac{1}{n+1} \sum_{k=0}^{n} m_k$$

our average-energy constraint amounts to minimize the limit

$$\liminf_{n\to\infty} \frac{1}{n+1} \sum_{k=0}^{n} \left(\sum_{j=0}^{k} m_j \right).$$

While the former is the average of all transition costs encountered along the run, the latter is the average of the accumulated cost (since the beginning of the run) along the run.

We proved that average-energy games have the same complexity as mean-payoff games (namely, in $\mathsf{NP} \cap \mathsf{coNP}$). We also studied the problem of combining energy- and average-energy constraints: our current findings is that in case the energy is requested to remain between given bounds, the problem is in $\mathsf{NEXPTIME} \cap \mathsf{coNEXPTIME}$. We also studied the one-player case (i.e., finding a run optimizing the average energy level while maintaining the energy level within given bounds), obtaining algorithms running in polynomial space.

- No technical report available yet.
- Partners: CNRS, UMONS and AAU.

5.2 Priced timed games

Building on the works we have published in CONCUR 2014 [BGK+14], we continue our study of *price timed games.* Those are zero-sum two-player games on priced timed automata. Depending on the ownership of one state, the players choose how long they stay in a state and which transitions they will take in order to minimize or maximize the cost of reaching target states. The cost itself is computed wrt to the edges traversed by the play and the time spent in each location. Thus, a price, which can be any rational number (positive or negative) is attached to each location and each transition.

In [ABM04] and [BCFL04], the authors propose simultaneously algorithms to solve those games with only positive prices and some constraints on the divergence of the cost of the runs. Those hypothesis were later justified in [BBR05] which show that in the general case, the problem of the synthesis of an optimal strategy is undecidable.

This undecidability result had prompted the study of the one-clock fragment of priced timed games: recently, Hansen *et al.* have proposed an algorithm to solve those games when all the rates are non-negative, and we have considered the case of bi-valued Priced Timed Games (where the prices of the locations can be only two values taken in some set $\{-p, 0, p\}$ for some rational number p) [BGK+14]

Our long-term goal is to obtain comprehensive results about the decidability status of one-clock priced timed games, in order to obtain *efficient algorithms for those problems.* In particular, we target iterative, fixed point algorithms (such as the value iteration techniques) that are known to perform well in many cases. We have recently made the following progresses:

- Nearly no properties are known about the value the minimiser (upper value) or the maximiser (lower value) can ensure in a PTG with both positive and negative costs. We have proved that the function associating the value of the game to the initial time valuation is continuous inside the regions, that we can give an upper and a lower bound on the value of all games unless this value is infinite, and that PTG are determined so the notion of upper value and lower value coincide.

- We have shown that the algorithm of [BGK+14] can easily be extended to solve a slightly larger class of games.

- We have shown that the algorithm introduced in [Rut11] for simple PTG can be adapted in order to work with both positive and negative costs. Thanks to some of the properties of the value proven previously and reductions of the problem, we obtain an exponential time algorithm for a broad class of PTG that we call negative-reset-acyclic. Those are the PTG whose cycles ending with a reset have positive cost or cost bounded away from 0.

- Finally, we have gained a better understanding of the difficulties of the problem and of the limits of other classical methods. In particular, we have proved that the value iteration algorithm [BCFL04] always *converges* to the value of the game but we have examples where it does not *terminate*.

- Technical report: Engel Lefaucheux's internship report.

- Partners: CNRS, ULB.

5.3 MITL synthesis and control

MITL is a real-time specification language which can express useful requirements such as 'every a event is eventually followed by a b event within at most 3 time units, and no sooner than 2 time units', written: $\Box(\mathtt{a} \implies \Diamond_{[2,3]}\mathtt{b})$. MITL has been introduced as an alternative to (actually, a strict subset) of MTL [Koy90] with better complexity [AFH96]: while MTL model-checking is, in general, undecidable [OW06], MITL formula can be translated to timed automata [AD94], thereby enabling automaton-based verification procedure. Recently, we have proposed new techniques based on alternating timed automata [BGE13, BEG14] that allow one to achieve more efficient implementations of model checking procedures for MITL.

While MITL seems well-suited for *verification* purpose, a natural extension towards control problems is highly desirable. In the untimed setting, the *realisability problem* has been well-studied for the sibling (non-real-time) logic LTL [Pnu77]. The input to this problem is a formula of LTL whose events are partitioned into controllable and uncontrollable ones. The problem then asks to compute automatically, if it exists, a reactive controller that enforces the given LTL requirement by controlling the controllable events, whatever the environment (controlling the uncontrollable events) does. This problem is known to be decidable in 2EXPTIME, and efficient implementations exist that can handle real-life specifications [FJR13].

Unfortunately, the realisability problem is undecidable for MITL [DGRR09], and this is due, roughly speaking, to the fact that, to enforce an MITL objective, a controller needs in general infinitely many clocks, with infinite precision. A natural sub-problem that is still meaningful in practice is to ask for the existence of a controller with *bounded resources* (number of clocks and precision). In [DM02], D'Souza *et al.* consider this problem in the setting of *infinite words*, and when the specification is given as a timed automaton. They provide an exponential-time algorithm, thereby leading a triply exponential time for MITL (since MITL can be translated to timed automata as recalled above). However, this procedure seems hard to implement because of the translation step to timed automata. On the other hand, Bouyer *et al.* consider in [BBC06] a very related problem of control of MTL specifications (still under bounded resources) in the *finite words* case. Since MITL is a syntactic fragment of MTL, their technique applies to MITL too. This technique is somehow *on-the-fly* and seems thus more likely to be implemented, but, unfortunately, its running time is non-primitive recursive, and it does not extend straightforwardly to infinite words.

We are currently considering the problem of control (in the sense of [DM02]) of MITL formula, with bounded resources, and we plan to apply to this problem the *efficient* techniques based on alternating automata we have developed for verification purpose [BGE13, BEG14]. With these techniques, we have already achieved the following progresses:

- Using our translation of MITL to alternating timed automata, we have defined an efficient version of Bouyer *et al.*'s algorithm [BBC06], custom tailored for MITL, that runs in triply exponential time (i.e., a better complexity than the non-primitive recursive algorithm for MTL, and matching with the complexity of D'Souza *et al.*'s procedure).

- We have defined several heuristics, in the spirit of antichains [WDMR08], that exploit the peculiar structure of the problem, and allow to speed up this algorithm.
- We have implemented the algorithm and the heuristics and applied them with success to some toy examples.

In the near future, we want to define efficient procedures to achieve control of MITL specifications in the infinite words case (assuming bounded resources) that are amenable to efficient implementation (unlike the general procedure of D'Souza *et al.*). To this end, we will extend the k-co-Büchi techniques of [DGGRS10].

- No technical report available yet. We expect to submit these results in the course of April.
- Partners: ULB, UMONS (in a collaboration with the Université Pierre et Marie Curie, Paris VI).

References

[ABM04] Rajeev Alur, Mikhail Bernadsky, and P. Madhusudan. Optimal reachability for weighted timed games. In *Proceedings of the 31st International Colloquium on Automata, Languages and Programming (ICALP'04)*, volume 3142 of *Lecture Notes in Computer Science*, pages 122–133. Springer, 2004.

[AD90] Rajeev Alur and David L. Dill. Automata for modeling real-time systems. In Mike Paterson, editor, *Proceedings of the 17th International Colloquium on Automata, Languages and Programming (ICALP'90)*, volume 443 of *Lecture Notes in Computer Science*, pages 322–335. Springer-Verlag, July 1990.

[AD94] Rajeev Alur and David L. Dill. A theory of timed automata. *Theor. Comput. Sci.*, 126(2):183–235, 1994.

[AFH96] Rajeev Alur, Tomás Feder, and Thomas A. Henzinger. The benefits of relaxing punctuality. *J. ACM*, 43(1):116–146, 1996.

[ALP01] Rajeev Alur, Salvatore La Torre, and George J. Pappas. Optimal paths in weighted timed automata. In Benedetto and Sangiovanni-Vincentelli [BSV01], pages 49–62.

[BBBR07] Patricia Bouyer, Thomas Brihaye, Véronique Bruyère, and Jean-François Raskin. On the optimal reachability problem of weighted timed automata. *Formal Methods in System Design*, 31(2):135–175, 2007.

[BBC06] Patricia Bouyer, Laura Bozzelli, and Fabrice Chevalier. Controller synthesis for MTL specifications. In *CONCUR 2006 - Concurrency Theory, 17th International Conference, CONCUR 2006, Bonn,* *Germany, August 27-30, 2006, Proceedings*, volume 4137 of *Lecture Notes in Computer Science*, pages 450–464. Springer, 2006.

[BBM06] Patricia Bouyer, Thomas Brihaye, and Nicolas Markey. Improved undecidability results on weighted timed automata. *Information Processing Letters*, 98(5):188–194, 2006.

[BBR05] Thomas Brihaye, Véronique Bruyère, and Jean-François Raskin. On optimal timed strategies. In *Proceedings of the Third international conference on Formal Modeling and Analysis of Timed Systems (FORMATS'05)*, volume 3829 of *Lecture Notes in Computer Science*, pages 49–64. Springer, 2005.

[BCFL04] Patricia Bouyer, Franck Cassez, Emmanuel Fleury, and Kim G. Larsen. Optimal strategies in priced timed game automata. In *Proceedings of the 24th Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS'04)*, volume 3328 of *Lecture Notes in Computer Science*, pages 148–160. Springer, 2004.

[BEG14] T. Brihaye, M. Estiévenart, and G. Geeraerts. On mitl and alternating timed automata over infinite words. In *Proceedings of FORMATS2014*, number 8711 in Lecture Notes in Computer Science, pages 69–84. Springer Verlag, 2014.

[BFH+01] Gerd Behrmann, Ansgar Fehnker, Thomas Hune, Kim Guldstrand Larsen, Paul Pettersson, Judi Romijn, and Frits W. Vaandrager. Minimum-cost reachability for priced timed automata. In Benedetto and Sangiovanni-Vincentelli [BSV01], pages 147–161.

[BGE13] T Brihaye, G. Geeraerts, and M. Estiévenart. On MITL and alternating timed automata. In *Proceedings of FORMATS 2013, 11th International Conference on Formal Modeling and Analysis of Timed Systems*, volume 8053 of *Lecture Notes in Computer Science*, pages 47–61. Springer, 2013.

[BGK+14] Thomas Brihaye, Gilles Geeraerts, Shankara Narayanan Krishna, Lakshmi Manasa, Benjamin Monmege, and Ashutosh Trivedi. Adding negative prices to priced timed games. In *CONCUR 2014 - Concurrency Theory - 25th International Conference, CONCUR 2014, Rome, Italy, September 2-5, 2014. Proceedings*, volume 8704 of *Lecture Notes in Computer Science*, pages 560–575. Springer, 2014.

[BJM14] Patricia Bouyer, Samy Jaziri, and Nicolas Markey. On the value problem in weighted timed games. Research Report LSV-14-12, Laboratoire Spécification et Vérification, ENS Cachan, France, October 2014. 24 pages.

[BLMR06a] Patricia Bouyer, Kim G. Larsen, Nicolas Markey, and Jacob Illum Rasmussen. Almost optimal strategies in one-clock priced timed games. In *Proceedings of the 26th Conference on Foundations of Software Technology and Theoretical Computer Science*

(FSTTCS'06), volume 4337 of *Lecture Notes in Computer Science*, pages 345–356. Springer, 2006.

[BLMR06b] Patricia Bouyer, Kim Gulstrand Larsen, Nicolas Markey, and Jacob Illum Rasmussen. Almost optimal strategies in one-clock priced timed automata. In S. Arun-Kumar and Naveen Garg, editors, *Proceedings of the 26th Conferentce on Foundations of Software Technology and Theoretical Computer Science (FSTTCS'06)*, volume 4337 of *Lecture Notes in Computer Science*, pages 345–356. Springer-Verlag, December 2006.

[BSV01] Maria Domenica Di Benedetto and Alberto L. Sangiovanni-Vincentelli, editors. *Hybrid Systems: Computation and Control, 4th International Workshop, HSCC 2001, Rome, Italy, March 28-30, 2001, Proceedings*, volume 2034 of *Lecture Notes in Computer Science*. Springer, 2001.

[DGGRS10] B. Di Giampaolo, G Geeraerts, J.F. Raskin, and N. Sznajder. Safraless procedures for timed specifications. In Springer, editor, *Proceedings of FORMATS 2010, 8th International Conference on Formal Modelling and Analysis of Timed Systems*, volume 6246 of *Lecture Notes in Computer Science*, pages 2–22, 2010.

[DGRR09] L. Doyen, G. Geeraerts, J.F. Raskin, and J. Reicher. Realizability of real-time logics. In *Proceedings of FORMATS 2009, 7th International Conference on Formal Modeling and Analysis of Timed Systems*, volume 5813 of *Lecture Notes in Computer Science*, pages 133–148. Springer, 2009.

[DM02] Deepak D'Souza and P. Madhusudan. Timed control synthesis for external specifications. In *STACS 2002, 19th Annual Symposium on Theoretical Aspects of Computer Science, Antibes - Juan les Pins, France, March 14-16, 2002, Proceedings*, volume 2285 of *Lecture Notes in Computer Science*, pages 571–582. Springer, 2002.

[FJR13] Emmanuel Filiot, Naiyong Jin, and Jean-François Raskin. Exploiting structure in LTL synthesis. *STTT*, 15(5-6):541–561, 2013.

[FKS14] Laurent Fribourg, Ulrich Kühne, and Romain Soulat. Finite controlled invariants for sampled switched systems. *Formal Methods in System Design*, 45(3):303–329, December 2014.

[HIJM13] Thomas Dueholm Hansen, Rasmus Ibsen-Jensen, and Peter Bro Miltersen. A faster algorithm for solving one-clock priced timed games. In *Proceedings of the 24th International Conference on Concurrency Theory (CONCUR'13)*, volume 8052 of *Lecture Notes in Computer Science*, pages 531–545. Springer, 2013.

[HIM13] Thomas Dueholm Hansen, Rasmus Ibsen-Jensen, and Peter Bro Miltersen. A faster algorithm for solving one-clock priced timed games. In Pedro R. D'Argenio and Hernán C. Melgratt, editors, *Proceedings of the 24th International Conference on Concurrency Theory (CONCUR'13)*, volume 8052 of *Lecture Notes in Computer Science*, pages 531–545. Springer-Verlag, August 2013.

[Koy90] Ron Koymans. Specifying real-time properties with metric temporal logic. *Real-Time Systems*, 2(4):255–299, 1990.

[Küh] Ulrich Kühne. Minimator Web page. `https://bitbucket.org/ukuehne/minimator/wiki/Home`.

[OW06] Joël Ouaknine and James Worrell. On metric temporal logic and faulty turing machines. In *Foundations of Software Science and Computation Structures, 9th International Conference, FOSSACS 2006, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2006, Vienna, Austria, March 25-31, 2006, Proceedings*, volume 3921 of *Lecture Notes in Computer Science*, pages 217–230. Springer, 2006.

[Pnu77] Amir Pnueli. The temporal logic of programs. In *Proceedings of FOCS'77*, pages 46–57. IEEE Computer Society Press, 1977.

[Rut11] Michał Rutkowski. Two-player reachability-price games on single-clock timed automata. In *Proceedings of the Ninth Workshop on Quantitative Aspects of Programming Languages (QAPL'11)*, volume 57 of *Electronic Proceedings in Theoretical Computer Science*, pages 31–46, 2011.

[WDMR08] Martin De Wulf, Laurent Doyen, Nicolas Maquet, and Jean-François Raskin. Antichains: Alternative algorithms for ltl satisfiability and model-checking. In *TACAS*, volume 4963 of *Lecture Notes in Computer Science*. Springer, 2008.