Automated Game Analysis via Probabilistic Model Checking: a case study

P. Ballarini [1], M. Fisher [2], M.J. Wooldridge [3]

Department of Computer Science, University of Liverpool, UK

Abstract

It has been recognised for some time that there are close links between the various logics developed for the analysis of multi-agent systems and the many game-theoretic models developed for the same purpose. In this paper, we contribute to this burgeoning body of work by showing how a probabilistic model checking tool can be used for the automated analysis of game-like multi-agent systems in which both agents and environments can act with uncertainty. Specifically, we show how a variation of the well-known alternating offers negotiation protocol of Rubinstein can be encoded as a model for the PRISM model checker. and its behaviour analysed through automatic verification of a number of probabilistic CTL's properties.

Key words: agents, uncertainty, probabilistic model-checking.

1 Introduction

In game theory, negotiation is thought of as a game where two players bargain over one or several items. The players' roles are distinct: the *seller* is willing to get a profit by selling the bargained items whereas the *buyer* is keen on spending his/her resources (money) to buy them. The *utility* of each player essentially depends on the game outcome (i.e. the value at which an agreement is reached). A player's strategy basically describes what his/her next move is going to be (i.e. in a monetary negotiation the value of subsequent offers a player is willing to make). The possible moves a player can make constitutes the set of *pure strategies* for him/her. If *uncertainty* is accounted for, then probability distributions are given to pure strategies and we talk about *mixed strategies*, hence *expected utilities*. Game-theory aims to study properties of the players' strategies. A strategy which maximises a player's utility

[1] Email: `paolo@csc.liv.ac.uk`

[2] Email: `michael@csc.liv.ac.uk`

[3] Email: `mike@csc.liv.ac.uk`

This is a preliminary version. The final version will be published in Electronic Notes in Theoretical Computer Science
URL: `www.elsevier.nl/locate/entcs`

(independently of the opponent's one) is said to be a *dominant* one, whereas a combination of strategies (for the two players) forms a *Nash equilibrium* if no player will get a higher utility by adopting another strategy given that the opponent will stick to the considered one. Given a game G, questions like, "is there any *dominant* strategy?", or, "is there any *Nash-equilibrium* strategy combination?", need to be addressed by researchers. In this paper we focus on a specific negotiation framework, namely the Rubinstein's alternating offers' one, and show how a probabilistic model-checker can be used as an alternative means for its analysis. We consider a limited number of strategy profiles and for each of them we determine the corresponding probability distribution over the set of *agreements*, hence the expected utility. This approach differs from both the simulation and mathematical analysis techniques usually employed for game analysis. In particular, our approach is automatic, as are simulation techniques, yet covers *all* possible behaviours of the system, as do mathematical analysis techniques. Thus, our approach provides an automatic way to analyse an exact computed expectation concerning the possible system behaviours.

The structure of this paper is as follows. In Section 2, we review the problem of negotiation, in particular the "alternating offers" protocol, and in Section 3 we outline the *model checking* approach, specifically *probabilistic model checking*. We bring these two aspects together in Sections 4 and 5, describing the probabilistic model of the bargaining protocol in Section 4 and carrying out a range of verification experiments on this scenario in Section 5. Finally, in Section 6, we provide brief concluding remarks.

2 The alternating-offers negotiation framework

A number of protocols and associated strategies for automatic negotiation in multi-agent systems have been developed over the past two decades. One of the earliest, and most influential, was the monotonic concession protocol and Zeuthen strategy adapted by Rosenschein and Zlotkin [10] from previous work by Harsanyi and Zeuthen. In that work the authors gave the first real evaluation of such negotiation approaches in *automated* negotiation settings. Most recent work in automated negotiation has focussed on an alternative model of negotiation: the *alternating offers* model proposed by Rubinstein [8]. Here we briefly describe it referring the interested reader to the literature for more details. In Rubinstein's offers model, agents take it in turns to make an action. The action can be either (i) put forward a proposal (offer), or (ii) accept the most recent proposal. We assume just two agents, a and $\hat{a}$, and negotiation takes place in a sequence of rounds, which we will assume are indexed by the natural numbers. Agent a begins, at round 0, by making a proposal x^0, which agent $\hat{a}$ can either accept (A) or reject (R). If the proposal is accepted, then the deal x^0 is implemented. Otherwise, negotiation moves to another round, where agent $\hat{a}$ makes a proposal (counter-offer) and agent

a chooses to either accept or reject it.

There is, of course, nothing to stop negotiation using the alternating offers protocol going on for ever: according to the protocol, they can just keep on Rejecting and Rejecting and Rejecting... If the agents *never* reach agreement, (i.e., a history of the proposal simply consists of a sequence of deals and associated Rs), then we define the outcome of negotiation to be the *conflict deal*, Θ. The following basic assumptions are made [8, p.33–34] [4] :

Disagreement is the worst outcome: Agents prefer any outcome at least as much as disagreement.

Agents seek to maximise utility: Agents really do prefer to get larger utility values.

Time is valuable: For any outcome x and times t_1 and t_2, both agents would prefer outcome x at time t_1 over outcome x at time t_2 if $t_2 > t_1$. Thus, given a choice between agreement x being made now and x being made later, we always prefer the former.

In Rubinstein's *alternating offers* framework, the tactic for generating players' proposals is a function of time (1). The offer made by agbent a to agent $\hat{a}$ (either the *buyer*, b or the *seller*, s) at time time t ($0 \leq t \leq T^a$, where T^a is player a's deadline) falls between its Initial-Price (IP^a) and its Reserved-Price (RP^a) and is defined as follows:

$$p^t_{a \to \hat{a}} = \begin{cases} IP^a + \phi^a(t)(RP^a - IP^a) & \textbf{for } a = b, \\ RP^a + (1 - \phi^a(t))(IP^a - RP^a) & \textbf{for } a = s, \end{cases} \tag{1}$$

where k^a determines the price to be offered by agent a in its first proposal. Several time-dependent functions can be characterised by means of the Negotiation Decision Function (NDF) $\phi^a(t)$, which is as follows:

$$\phi^a(t) = k^a + (1 - k^a)(\frac{t}{T^a})^{\frac{1}{\psi}} \tag{2}$$

By varying the value of ψ in (2), different type of tactics can be obtained (Figure 1): with $\psi < 1$ we talk about *boulware* tactics (offer increases/decreases very slowly for most of the time and reaches the RP rapidly as the time deadline approaches), with $\psi = 1$, we have *linear* tactics whereas with $\psi > 1$ we talk about *conceder* strategies.

3 Probabilistic Model Checking

Model Checking [4] is a well established methodology for testing a system's model against properties expressed in terms of some temporal logic formulae. A *model checker* takes a model $\mathcal{M}$ and a formula ϕ as inputs and returns either YES if ϕ is satisfied on all executions through $\mathcal{M}$ (i.e. $\mathcal{M} \models \phi$) or NO

[4] Some other technical assumptions are made, but these are the "central" ones.

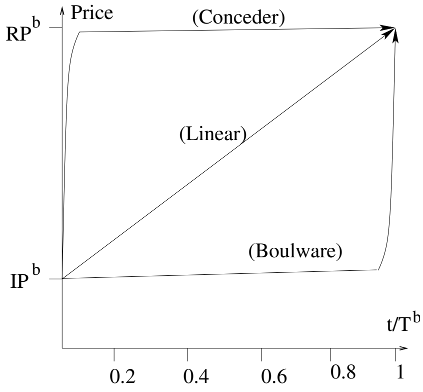


Fig. 1. Negotiation Decision Function

if it is not (i.e. $\mathcal{M} \not\models \phi$), providing, in such a case, a counter-example of the checked property.

Model-checking techniques may be classified with respect to the type of model they refer to. In this sense we may distinguish between models which allow to represent *non-determinism* without enclosing any "quantification" of the *uncertainty* (i.e. *non-probabilistic* system), as opposed to models which incorporate information about the likelihood of possible future evolutions (i.e. *probabilistic* systems). *Non-probabilistic* systems can be modelled in terms of Labelled Transition Systems (LTS), essentially state-graphs whose nodes are attached with *propositions* stating what is true in a state. Linear Time-temporal Logic [9] and its *branching-time* extension, the Computational Tree Logic [3], allow for the verification of *qualitative properties* against a LTS (e.g. properties such as *"in any possible execution a safe-state is reached at some point"* or *"no deadlock-state can be ever reached"*). When indications about the likelihood of the system behaviour can be devised, then a *probabilistic model* may be built. Markov processes [5] are a subclass of stochastic processes suitable for modelling systems such that the probability of possible future evolutions depends uniquely on the current state rather than on its *past history* (i.e. the path which lead to it). A system's timing is also taken care of with Markov chains models leading to either Discrete time Markov chains (DTMC), for which time is considered as discrete quantity, or Continuous time Markov chains (CTMC), when time is thought as a continuous one. The verification of Markov chains via model checking has been widely developed during last decades, resulting in the characterisation of specific temporal logics and verification algorithms: the Probabilistic Computational Tree Logic (PCTL) [6] for verification of DTMC and the Continuous Stochastic Logic (CSL) [2], for CTMC verification. In the following we briefly introduce the basic for PCTL model checking, which is the verification technique we are referring to in this work. For a detailed treatment the reader is referred to the literature.

Definition 3.1 *Given a set of atomic propositions AP, a labelled DTMC $\mathcal{M}$ is a tuple $(S, \mathbf{P}, L)$ where S is a finite set of states, $\mathbf{P} : S \times S \rightarrow [0, 1]$ is* *the* transition probability *matrix such that* $\forall s \in S$, $\sum_{s' \in S} \mathbf{P}(s, s') = 1$ *and* $L : S \to 2^{AP}$ *is a labelling function.*

A *path* in a given a DTMC $\mathcal{M} = (S, \mathbf{P}, L)$ and its *probability measure* are formally characterised in the following definitions.

Definition 3.2 *A path* σ *from state* s_0 *is an infinite sequence* $\sigma = s_0 \to s_1 \to \ldots \to s_n \to \ldots$ *such that* $\forall i \in \mathbb{N}$, $\mathbf{P}(s_i, s_{i+1}) > 0$*. Given* σ, $\sigma[k]$ *denotes the* k*-th element of* σ*.*

The probability measure of a set of infinite paths with common finite prefix $\sigma \uparrow n = s_0 \to \ldots \to s_n$ is defined as the product of the probability of the transitions their common prefix consists of.

Definition 3.3 *Let* $\sigma \uparrow n = s_0 \to \ldots \to s_n$ *be a finite path of* $\mathcal{M}$*. The probability measure of the set of (infinite) paths prefixed by* $\sigma \uparrow n$ *is*

$$Prob(\sigma \uparrow n) = \prod_{i=0}^{n-1} \mathbf{P}(s_i, s_{i+1})$$

if $n > 0$*, whereas* $Prob(\sigma \uparrow n) = 1$ *if* $n = 0$*.*

Definition 3.4 (PCTL syntax) *The syntax of PCTL state-formulae* (ϕ) *and path-formulae* (φ) *is inductively defined as follows with respect to the set of atomic propositions* AP*:*

$$\begin{aligned} &\phi := a \mid tt \mid \neg\phi \mid \ \phi \wedge \phi \mid \mathcal{P}_{\trianglelefteq p}(\varphi) \\ &\varphi := \phi \ U^{\leq t} \ \phi \end{aligned}$$

where $a \in AP$*,* $p \in [0,1]$*,* $t \in \mathbb{N}^* \cup \{\infty\}$ *and* $\trianglelefteq \in \{\geq, >, \leq, <\}$*,*

The PCTL semantics is as the CTL one except for probabilistic path-formulae. $\mathcal{P}_{\trianglelefteq p}(\phi' \ U^{\leq t} \phi'')$ is satisfied in a state s iff the probability measure of paths starting at s and satisfying $(\phi' \ U^{\leq t} \phi'')$ fulfils the bound $\trianglelefteq p$. Formally:

$$s \models \mathcal{P}_{\trianglelefteq p}(\phi' \ U^{\leq t} \phi'') \ \ \textbf{iff} \ \ Prob(s, (\phi' \ U^{\leq t} \phi'')) \trianglelefteq p$$

where the semantics of $(\phi' \ U^{\leq t} \phi'')$ with respect to a path σ is defined as:

$$\sigma \models \phi' U^{\leq t} \phi'' \ \ \textbf{iff} \ \ \exists i \leq t : \sigma[i] \models \phi'' \wedge \forall j < i, \sigma[j] \models \phi'$$

Essentially PCTL extends CTL's expressiveness in two ways: by allowing a continuous *path-quantification* (i.e. CTL *existential* and *universal* path quantifiers are replaced by a single *continuous* quantifier, namely $\mathcal{P}_{\trianglelefteq p}$) [5] and by introducing a *discrete* time-bounding for Until-formulae. For a complete treatment of PCTL model checking we refer the reader to [6].

[5] PCTL is a superset of CTL's as: $E(\phi U \phi) \equiv \mathcal{P}_{>0}(\phi U \phi)$ and $A(\phi U \phi) \equiv \mathcal{P}_{\geq 1}(\phi U \phi)$

4 DTMC Model of Rubintein's Protocol

In this section we describe how we have built the DTMC model for the negotiation framework introduced in Section 2. The model has been implemented in the *Reactive Modules* formalism of Alur and Henziger [1], the input language for the Probabilistic Model-Checker PRISM [7], and it consists of two modules, the *Buyer* and the *Seller*, that reproduce the players' behaviour described by the UML state-chart diagrams of Figure 2. In essence the *buyer* and the *seller* alternatively throw a *bid/counter_bid* then wait for the other player to make a decision on their offer. If the offer is accepted (i.e. an agreement has been reached) then the purchase takes place and the negotiation is successfully completed. State-diagrams in Figure 2 refer to a configuration where the *Buyer* is starts the bidding process. In order to assess symmetry of the bargaining process both starting orders (i.e. the *buyer* bids first or the *seller*) must be considered.

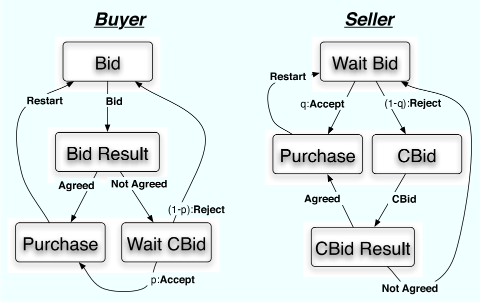


Fig. 2. Buyer and Seller state-charts

For a player the decision on the opponent's offer is a probabilistic one, which depends on the offered amount. Such a probability is formally given by the Seller and Buyer Acceptance Probability functions,$S\_AP()$ and $B\_AP()$ respectively, which we have defined in the following manner:

$$S\_AP(x) = \begin{cases} 0 & \textbf{if } x <= S\_RP \\ 1 - \frac{S\_RP}{x} & \textbf{if } x > S\_RP \end{cases} \qquad (3)$$

$$B\_AP(x) = \begin{cases} 1 & \textbf{if } x <= 0 \\ 1 + \frac{S\_RP}{x - (B\_RP + S\_RP)} & \textbf{if } S\_RP < x < B\_RP \\ 0 & \textbf{if } x > B\_RP \end{cases} \qquad (4)$$

In defining such functions two aspects must be accounted for. First, offers ruled out by a player's *reserved price* must be given a null probability. Secondly, for the sake of symmetry, players should be equally likely to accept offers of *equally utility* (i.e. whose distance from their respective RP is the same). This can straightforwardly be verified on Figure 3 where curves for

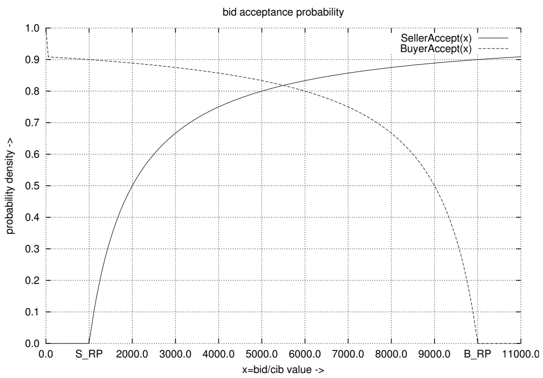


Fig. 3. Offer's acceptance probability

both function 3 and 4 are shown[6].

Piecewise approximation of NDFs. In our model the continuous NDFs ($\phi(t)$) belonging to the family defined in equation 2 are approximated by piecewise linear functions consisting of two pieces[7] (see Figure 4)[8]. The offer function has three parameters: the slope of the first piece, the slope of the second piece and boundary (switch time) between the pieces. The desired setting (boulware/conceder strategy) is chosen through model configuration so that different *strategy profiles* (strategy combinations) are verified (i.e. probability for each possible negotiation outcome are derived).

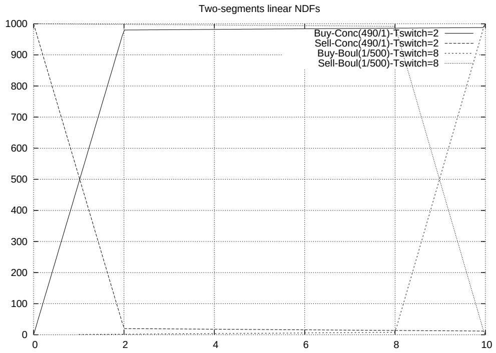


Fig. 4. NDFs' linear approximation

[6] the depicted curves correspond to a setting such that the seller and buyer reserved price are, respectively, set to $S\_RP = 1000$ and $S\_RP = 10000$.

[7] A two piece line is a good approximation for *extreme* bargaining tactics (i.e. $\psi \sim 0$ or $\psi >> 1$), which is the type of non-linear tactics we are interested to address in this work. Less extreme strategies may be better approximated by multi-piece lines, which would require minimal modifications to our model in order to be coped with.

[8] the depicted curves correspond to specific settings for the pieces' slopes and switch-time.

5 Model's verification

The probabilistic behaviour (i.e. the Acceptance probability functions) encoded within the DTMC model of the alternating offer framework, results in a probability distribution over the set of possible outcomes of the negotiation (i.e. the interval $[S\_RP, B\_RP]$), hence over a strategy profile's payoff. Once a strategy profile (e.g. Boulware-Boulware or Conceder-Conceder, etc.) has been chosen by configuration, the corresponding probability distribution is determined by verifying, with the PRISM tool, the following PCTL formula against the framework model:

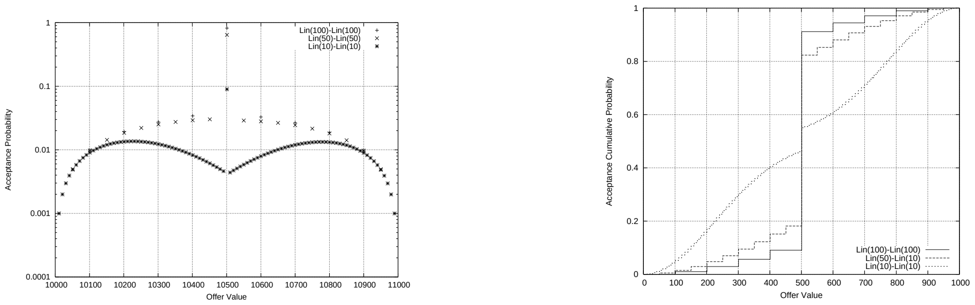
$$P\!=?(tt\ U(agreement) \wedge (purchase\!=\!PVAL)) \tag{5}$$

The above property captures those evolutions for which an *agreement* over a specific value ($PVAL$) is reached, at some point, during the bargaining process. The verification of (5) for every possible agreement value ($PVAL \in [S\_RP, B\_RP]$) provide us with the distribution of the probability over the set of outcomes. From this the resulting expected utility can be straightforwardly derived from the verification results (for the sake of brevity we do not report it in here).

Before starting the verification phase the model needs to be configured. The configuration requires setting a number of parameters amongst which the strategy combination which we want to study (i.e. the strategy's slopes and switch-time for both players'). The possible strategies are clearly infinite, however for our purpose we consider only a limited number of combinations which we aim to compare through the results of model verification. The range of possible *agreements* (i.e. the interval $[S\_RP, B\_RP]$) must also be chosen through configuration and, in order to improve the model's efficiency, this is done by means of two distinct parameters: the interval width and its offset from the origin. We observe that the numerical result of model verification is affected by the chosen interval (as a result of functions (3) and (4) definition).

In the following we report some of the results we have obtained by verification of (5) for an accepting interval arbitrary set to $[10000, 11000]$ (i.e. width:10^2, offset;10^3). We will compare the resulting probability distribution (cumulative) for different strategy profiles, pointing out which strategy is dominant amongst the considered ones. The results we present are grouped according to the family the underlying strategy combination belongs to. Hence, for example, $Lin(50)$-$Lin(50)$, denotes a profile in which both players are adopting a (pure) linear offer's function with slope equal to 50 whereas, for example, in $Boul(1/100)$-$Conc(100/1)$ the seller is adopting a boulware strategy with slopes equal to 1 and 100 respectively, while the buyer is conceding with an initial gradient of 100 which becomes equal to 1 after switching.

Linear-Linear symmetrical: in this setting we consider strategy profiles in which both players are using a linear offer's function with equal slope. In Figure 5(a) the probability distribution for the possible agreements is compared for different value of the strategy slope (i.e. $Lin(10)$, $Lin(50)$ and $Lin(100)$). This graph essentially shows that larger slope's values result in a higher probability for equal values in the accepting interval. The corresponding expectation values are roughly the same: $Exp(Lin(10)) \sim 498$, $Exp(Lin(50)) \sim 499$ and $Exp(Lin(100)) \sim 500$, showing a larger slope tend to advantage the seller. This is also confirmed by the graphs in Figure 5(b) that represent the cumulative distribution functions for the three cases. The curves in there show that larger strategy's gradients are better from the seller point of view as they concentrate most of the probability close to the supremum of the accepting interval.



(a) Probability distribution

(b) Cumulative distribution

Fig. 5. Linear-Linear strategy profile

Linear-Linear asymmetrical: here we consider strategy profiles of linear offer's functions with different slopes. For studying the incidence of the strategy's gradient on the probabilistic outcome of negotiation we consider a fixed value for the slope of one player while varying the other's. In Figure 6 the cumulative distributions for several asymmetrical linear profiles are depicted. We observe that by increasing the buyer's slope while the seller's one is set, for example, to 10 (e.g. by comparing the curves for profiles $Lin(50)$-$Lin(10)$, and $Lin(100)$-$Lin(10)$) the probability tend to cumulate closer to the supremum of the interval (which is good for the seller). Again this is confirmed by looking at the expectation values, which show the same tendency, with $Exp(Lin(50)Lin(10)) \sim 692$, whereas $Exp(Lin(100)Lin(10)) \sim 804$.

Non-linear asymmetrical: similar conclusions are valid also when we consider profiles for which both players are using a non-linear strategy with different slope values. In Figure 7 the cumulative probability for asymmetrical Conceder combinations with different gradients are compared. There we can observe that profiles which switch to the *low gradient* piece of the strategy

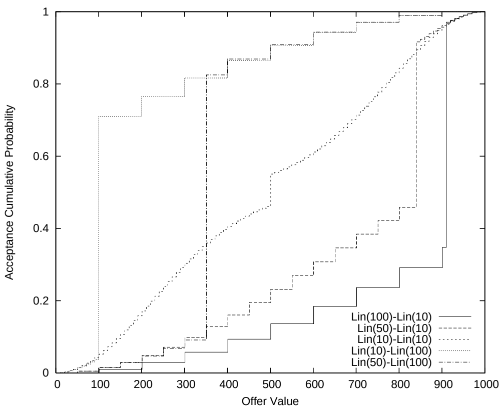


Fig. 6. Lin-Lin asymmetrical cumulative

earlier tend to cumulate the probability closer to the *higher utility* half of the interval. This is evident, for example, if we compare curves of equal gradients but different swith time as in $Conc(10/1)$-$Conc(100/10)$-$T\_sw : 4$ and $Conc(10/1)$-$Conc(100/10)$-$T\_sw : 8$ in Figure 7, the former corresponding to an earlier switch time for the seller (equal to 4) the latter to a delayed switch time (equal to 8).

Similar considerations hold when combinations of different type of non-linear strategies, are compared. In Figure 8 the cumulative probability for boulware-conceder profiles of fixed gradients (i.e. respectively $Boul(1/100)$ and $Conc(100/1)$) are compared with respect to to different value of switch time for the seller. The graphs in Figure 8 confirm that for constant slope's values, a non-linear strategy with the earliest switch time is dominant.

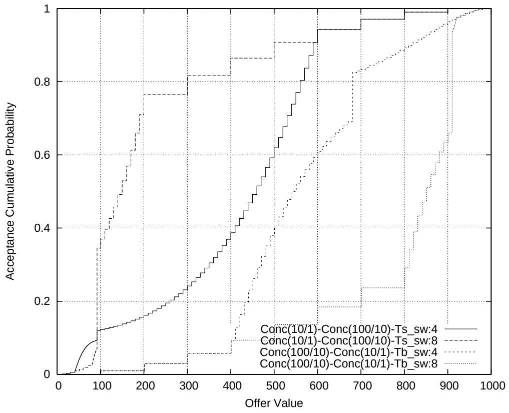


Fig. 7. Conc-Conc asymmetrical cumulative

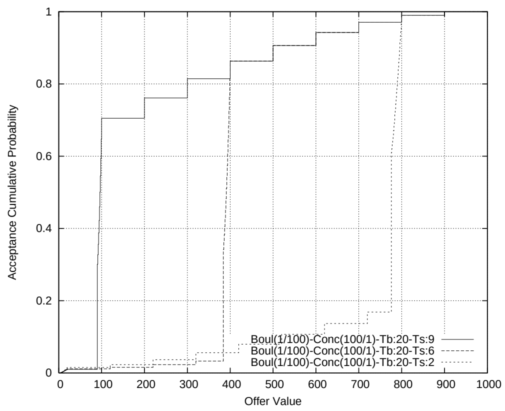


Fig. 8. Boul-Conc asymmetrical cumulative

6 Conclusion

In this paper we have shown a different approach to the verification of multi-player games which can be used as an alternative to (or in conjunction with) analytical methods and/or simulation. We have illustrated how the analysis of specific negotiation (mixed) strategies can be performed by means of probabilistic model checking. This is achieved by developing an *ad hoc* probabilistic model which is then verified against probabilistic properties.

This analysis has helped in comparing the effect that several strategic variables has on the probabilistic outcome of negotiation. In particular we have shown that, for large value of the offers functions slope (in both linear and non-linear strategies) the expected value tends to increase and the probability distribution tends to cumulate toward the supremum of the accepting interval. Furthermore with respect to non-linear strategies of constant slopes, the earliest switch time (to the low gradient piece of piecewise linear offer function) is *dominant* with respect to the others.

While we have only presented a selection of the results obtained, it is clear that this approach can be used to provide a deep analysis of probabilistic negotiation/game scenarios. Consequently, we believe this approach has very great potential for improvements in game analysis, in automated analysis procedures, and in the development of sophisticated negotiation strategies.

Acknowledgement: This work was supported by an EC Marie Curie Fellowship under contract number HPMF-CT-2001-00065.

References

[1] R. Alur and T. Henzinger. Reactive modules. *Formal Methods in System Design*, 15(1):7–48, 1999.

[2] C. Baier, B. Haverkort, H. Hermann, and J.P. Katoen. Model-checking algorithms for continuous-time Markov chains. *IEEE Trans. on Software Eng.*, Vol. 29(6):pp. 524–541, June 2003.

[3] E. Clarke, E. Emerson, and A. Sistla. Automatic verification of finite-state concurrent systems using temporal logic specifications. *ACM Transactions on Programming Languages and Systems*, 8(2):244–263, 1986.

[4] E. M. Clarke, O. Grumberg, and D. A. Peled. *Model Checking.* MIT Press, 2000.

[5] W. Feller. *An introduction to probability theory and its applications.* John Wiley and Sons, 1968.

[6] H. A. Hansson and B. Jonsson. A framework for reasoning about time and reliability. In *Proc. 10th IEEE Real -Time Systems Symposium*, pages 102–111, Santa Monica, Ca., 1989. IEEE Computer Society Press.

[7] M. Kwiatokowska, G. Norman, and D. Parker. Probabilistic symbolic model checking with prism: A hybrid approach. *International Journal on Software Tools for Technology Transfer (STTT)*, 2004.

[8] M. J. Osborne and A. Rubinstein. *Bargaining and Markets.* The Academic Press: London, England, 1990.

[9] A. Pnueli. A temporal logig of programs. *Proc. of the 18th IEEE Symposioum on Foundations of Computer Science.*, pages 46–57, 1977.

[10] J. S. Rosenschein and G. Zlotkin. *Rules of Encounter: Designing Conventions for Automated Negotiation among Computers.* The MIT Press: Cambridge, MA, 1994.