

# The Robustness of Periodic Orchestrations in Uncertain Evolving Environments <sup>\*</sup>

J. Castro<sup>1</sup>, J. Gabarro<sup>2</sup>, M. Serna<sup>2</sup>, A. Stewart<sup>3</sup>

<sup>1</sup> LARCA Research group. Computer Science Dept. UPC Barcelona, Spain

<sup>2</sup> ALBCOM Research group. Computer Science Dept. UPC Barcelona, Spain

<sup>3</sup> School of Computer Science. The Queen's University of Belfast, UK

**Abstract.** A framework for assessing the robustness of long-duration *repetitive orchestrations* in uncertain *evolving* environments is proposed. The model assumes that service-based evaluation environments are stable over *short* time-frames only; over longer periods service-based environments evolve as demand fluctuates and contention for shared resources varies. The behaviour of a short-duration orchestration  $E$  in a stable environment is assessed by an *uncertainty profile*  $\mathcal{U}$  and a corresponding zero-sum angel-daemon game  $\Gamma(\mathcal{U})$  [2]. Here the angel-daemon approach is extended to assess *evolving* environments by means of a subfamily of *stochastic games*. These games are called *strategy oblivious* because their transition probabilities are strategy independent. It is shown that the value of a strategy oblivious stochastic game is well defined and that it can be computed by solving a linear system. Finally, the proposed stochastic framework is used to assess the evolution of the Gabrnm IT system.

**Keywords.** Orchestrations, uncertainty, zero-sum games, stochastic games.

## 1 Introduction

Web services pervade modern life; commonplace examples include media and health-care services. Complex applications can be (rapidly) built by interconnecting (or orchestrating) a set of underlying services. Load balancing, provisioning, and failure modes are fundamental issues for the cloud computing community; during an execution of an application the performance of *some* component services may degrade because of over-demand. However, cloud providers try to balance work-loads across computing resources. The overall performance of an application is affected by the interplay between positive and negative (competing) environmental influences. One approach to making service-based systems resilient is to use *ad hoc* techniques, based on the wisdom and folklore of experienced engineers. The goal of this paper is use formal methods to reason about the resilience of long duration service-based systems.

---

<sup>\*</sup> J. Gabarro and M. Serna are partially supported by funds from the Spanish Ministry for Economy and Competitiveness (MINECO) and the European Union (FEDER funds) under grant TIN2013-46181-C2-1-R (COMMAS) and also by SGR 2014:1137 (ALBCOM) from AGAUR, Generalitat de Catalunya. J. Castro is partially supported by the Spanish Ministry for Economy and Competitiveness (MINECO) and the European Union (FEDER funds) under grants TIN2011-27479-C04-03 (BASMATI) and TIN2014-57226-P (APCOM) and by SGR 2014:890 (MACDA) from AGAUR, Generalitat de Catalunya.

In [2] orchestration games are constructed, with one player (the daemon) maliciously degrading a bounded number of services (to cause the maximum delay) and the other player (the angel) applying bounded elasticity to improve performance. Uncertainty profiles and strategic situations are used in [2] to characterise *stable* evaluation environments. Nash equilibria are used to characterise the performance and resilience of applications when subject to complex environmental influences. In this paper game theory is applied to more complex scenarios where patterns of environmental stress evolve throughout an application’s execution.

In order to analyse a long duration application which *repeatedly* evaluates an orchestration  $E$  we propose an extended stochastic uncertainty model. In the model execution environments are assumed to remain stable during evaluations of  $E$ . However, the environment may evolve between any of the periodic evaluations because of fluctuations in demand. It is assumed that the number of evaluation environments is finite and that evolution follows a Markovian process. Under this hypothesis the evolution of the environment can be assessed by means of stochastic games in which the future is described by means of a lottery. Stochastic games [8] have been widely used to study the inter-temporal behaviour of “real” systems [3]. In a zero-sum stochastic game  $\Gamma = \langle \Gamma^1, \dots, \Gamma^\ell \rangle$  each state  $l$  is formed by two components a zero sum game and a lottery. In state  $l$ , a player engages in the zero-sum game and after moves probabilistically to the next state.

For analyzing periodic orchestrations, it seems sufficient to consider oblivious lotteries, where the probability of changing state depends only on the current state. We prove that (i) games in the family of (zero-sum) *strategy oblivious* stochastic games have a well defined value, in the discounted model, and (ii) a game value can be computed by solving a linear system. This result allows the proposed framework to be applied to analyse the behaviour of the **Gabrmn** system.

The paper is organised as follows. Section 2 introduces periodic orchestrations while Section 3 introduces zero-sum, stochastic and *strategy oblivious* games. Section 3 also provides techniques for assessing complex evolving scenarios (Theorem 2). In Section 4 we discuss the assessment of both short- and long-duration orchestrations. The behaviour of the **Gabrmn** system in an evolving environment is analysed in Section 5. Finally, we draw some conclusions in Section 6.

## 2 Periodic Orchestrations

The language **Orc** [5, 4] can be used to model the co-ordination of a set of loosely-coupled services. **Orc** has a well defined semantics [10, 12] and so lends itself to the study of orchestration behaviour.

The simplest **Orc** expression is a service call. When called a service responds by publishing a result (for example a link to a web page). Three predefined services (or internal sites) are used in this paper: 0, 1 and  $Rtimer(\tau)$ . Service 0 never publishes; a call to  $1(x)$  echoes back its input argument  $x$ ; a call to  $Rtimer(\tau)$  publishes a result after  $t$  time steps. Any two orchestrations  $P$  and  $Q$  can be composed using the operators:

- **Sequential composition**  $P > x > Q(x)$ : Initially  $P$  is evaluated: for each output  $v$ , published by  $P$ , an instance  $Q(v)$  is invoked. If  $P$  publishes the stream of values,

$v_1, v_2, \dots, v_n$ , then orchestration  $P > x > Q(x)$  publishes some interleaving of the set  $\{Q(v_1), Q(v_2), \dots, Q(v_n)\}$ . The abbreviation  $P \gg Q$  is used in situations where  $Q$  does not depend on  $x$ .

- Parallel composition  $P \mid Q$ : The independent orchestrations  $P$  and  $Q$  are executed in parallel;  $P \mid Q$  publishes *some* interleaving of the values published by  $P$  and  $Q$ .
- Pruning  $P(x) < x < Q$ : Orchestrations  $P$  and  $Q$  are evaluated in parallel;  $P$  may become blocked by a dependency on  $x$ . The first result published by  $Q$  is bound to  $x$ , the remainder of  $Q$ 's evaluation is terminated and evaluation of the blocked residual of  $P$  is resumed.

Consider the periodic computation  $D = (E \mid (Rtimer(\tau) \gg D))$  which repeatedly calls a short-duration orchestration  $E$ , say  $E = (A \mid B) < x < (F \mid G)$ , at time intervals  $\tau$ . It is assumed that the environment of  $E$  remains stable during any evaluation. During a particular evaluation of  $E$  some underlying services may be degraded, because of excessive demand, while other services may benefit from environmental resilience. The precise nature of the environmental factors in play at any one moment in time is difficult to characterise in a quantitative way. A qualitative environmental characterisation can be given using an uncertainty profile [2]. This specifies potential positive and negative environmental influences (e.g. overdemand, elasticity). The combined effect of these influences on an evaluation of  $E$  can be assessed using game theory [2]. For a periodic evaluation the environmental influences may evolve from one time period to the next. The goal of this paper is to demonstrate how stochastic games can be used to analyse such evolving situations.

### 3 Preliminaries on Games

Zero-sum and stochastic zero-sum games are introduced below; Standard notation is used throughout:  $\Delta(S)$  denotes the set of probability distributions over a finite set  $S$ . Zero-sum games can be used to model stable stressed environments.

**Definition 1.** A two player zero-sum game is a strategic game described by the tuple  $\Gamma = \langle A_1, A_2, u \rangle$ .  $A_1$  and  $A_2$  are the set of eligible actions for player 1 and 2, respectively. The third component is a mapping  $u$  from  $A_1 \times A_2$  to the rational numbers. An element  $(i, j) \in A_1 \times A_2$  is called a strategy profile. The utility of strategy profile  $(i, j)$  is  $u(i, j)$ , for player 1, and  $-u(i, j)$ , for player 2. Utilities are rational numbers.

A mixed strategy is a lottery on the set of eligible actions. Utility  $u$  can be extended over mixed strategy profiles. Given a mixed strategy profile  $(\alpha, \beta) \in \Delta(A_1) \times \Delta(A_2)$  where  $\alpha = (\alpha_1, \dots, \alpha_n)$  and  $\beta = (\beta_1, \dots, \beta_m)$  then  $u(\alpha, \beta) = \sum_{i,j} \alpha_i u(i, j) \beta_j$ . The value of a zero-sum game  $\Gamma$  is defined as  $\nu(\Gamma) = \max_{\alpha \in \Delta(A_1)} \min_{\beta \in \Delta(A_2)} u(\alpha, \beta)$ . For any (mixed) Nash equilibrium  $(\alpha, \beta)$  it is known that  $u(\alpha, \beta) = \nu(\Gamma)$ .

*Example 1.* The class of  $2 \times 2$ -zero sum games is well-known [6]. Consider games  $\Gamma = (\{T, B\}, \{L, R\}, u)$  and  $\Gamma' = (\{T, B\}, \{L, R\}, u')$  below with utilities  $u$  and  $u'$ .

|   |     |     |
|---|-----|-----|
|   | L   | R   |
| T | 1/2 | 1   |
| B | 1   | 1/2 |

$\Gamma$

|   |     |     |
|---|-----|-----|
|   | L   | R   |
| T | 3   | 3/2 |
| B | 3/2 | 3   |

$\Gamma'$

In a Nash equilibrium with full support  $u(T, \beta) = u(B, \beta) = \nu(\Gamma)$  and  $u(\alpha, L) = u(\alpha, R) = \nu(\Gamma)$ . Thus  $\alpha = \beta = 1/2$ ,  $\nu(\Gamma) = 3/4$  and  $\nu(\Gamma') = 9/4$   $\square$

The following linear transformation result for zero-sum games is well-known:

**Lemma 1.** *Let  $\Gamma = (A_1, A_2, u)$  and  $\Gamma' = (A_1, A_2, u')$  be two zero-sum games with  $u'(i, j) = au(i, j) + b$ , for some  $a > 0$ . Then games  $\Gamma$  and  $\Gamma'$  have the same set of Nash equilibria and  $\nu(\Gamma') = a\nu(\Gamma) + b$ .*

Following [6, 9] stochastic games are formally defined as:

**Definition 2.** *A two person stochastic game  $\Gamma = \langle \Gamma^1, \dots, \Gamma^\ell \rangle$  is a tuple of  $\ell$  sub-games. Each sub-game (or state)  $\Gamma^l$  has form  $\Gamma^l = \langle \mathbf{game}^l(\Gamma), \mathbf{lotteries}^l(\Gamma) \rangle$ . Here  $\mathbf{game}^l(\Gamma)$  is a zero-sum game and  $\mathbf{lotteries}^l(\Gamma)$  determines the next game:*

- $\mathbf{game}^l(\Gamma) = \langle A_1^l, A_2^l, u^l \rangle$  where  $A_1^l = \{1, \dots, n_l\}$ ,  $A_2^l = \{1, \dots, m_l\}$ , and
- $\mathbf{lotteries}^l(\Gamma) : A_1^l \times A_2^l \rightarrow \Delta(\{1, \dots, \ell\})$ .

When appropriate  $\mathbf{game}^l(\Gamma)$  and  $\mathbf{lotteries}^l(\Gamma)$  can be abbreviated to  $\mathbf{game}^l$  and  $\mathbf{lotteries}^l$ .

*Example 2.* A stochastic game  $\Gamma$  can be represented by a bi-matrix for each state combining utility and lottery. For instance,

|   |                 |                 |
|---|-----------------|-----------------|
|   | L               | R               |
| T | 1/2, (1/3, 2/3) | 1, (2/3, 1/3)   |
| B | 1, (2/3, 1/3)   | 1/2, (1/3, 2/3) |

$\Gamma^1$

|   |                 |                 |
|---|-----------------|-----------------|
|   | L               | R               |
| T | 3, (1/4, 3/4)   | 3/2, (3/4, 1/4) |
| B | 3/2, (3/4, 1/4) | 3, (1/4, 3/4)   |

$\Gamma^2$

represents  $\Gamma = \langle \Gamma^1, \Gamma^2 \rangle$  where  $\mathbf{game}^1 = \Gamma$  and  $\mathbf{game}^2 = \Gamma'$  are the games in Example 1 and, for instance,  $\mathbf{lotteries}^1(\{T, L\}) = (1/3, 2/3)$ .  $\square$

A stochastic game is played through so called *stationary strategies* [8]. A *stationary strategy* is a pair  $(\alpha, \beta)$  formed by  $\alpha = (\alpha^1, \dots, \alpha^\ell)$  and  $\beta = (\beta^1, \dots, \beta^\ell)$  where, for any  $1 \leq l \leq \ell$ ,  $\alpha^l \in \Delta(A_1^l)$  and  $\beta^l \in \Delta(A_2^l)$ . Thus a stationary strategy comprises a mixed strategy profile for each game state. Lotteries are extended on mixed strategies as  $\mathbf{lotteries}_k^l(\alpha^l, \beta^l) = \sum_{i,j} \alpha_i^l \mathbf{lotteries}_k^l(i, j) \beta_j^l$  where  $\mathbf{lotteries}_k^l$  denotes the  $k$  component of  $\mathbf{lotteries}^l$ .

*Example 3.* A stationary strategy, for the game  $\Gamma$  given in Example 2, where player 1 selects  $T$  in  $\mathbf{game}^1$  and  $B$  in  $\mathbf{game}^2$  while player 2 chooses  $L$  in  $\mathbf{game}^1$  and  $R$  in  $\mathbf{game}^2$ , is  $((\alpha^1, \alpha^2), (\beta^1, \beta^2)) = (((1, 0), (0, 1)), ((1, 0), (0, 1)))$   $\square$

A stochastic game defines a collection of never ending games, one for each initial (sub-game) state. In a game players are rewarded and use a joint lottery to determine the next state. The  $\lambda$ -discounted reward model,  $0 < \lambda < 1$ , is used to define a utility for a stationary strategy. Consider the game with initial state  $\Gamma^l$  and with a stationary strategy  $(\alpha, \beta)$ . In the  $\lambda$ -discounted reward model the total payoff for player 1 is computed solving  $\mathfrak{P}_\lambda^l(\alpha, \beta) = \lambda u^l(\alpha^l, \beta^l) + (1 - \lambda) \sum_k \text{lotteries}_k^l(\alpha^l, \beta^l) \mathfrak{P}_\lambda^k(\alpha, \beta)$ . In the following  $\Gamma[\lambda]$  denotes the stochastic game  $\Gamma$  with a discount factor  $\lambda$ .

*Example 4.* Consider the stochastic game  $\Gamma[\lambda]$  with stationary strategy  $(\alpha, \beta)$  from Example 3 where the game starts in state 1. Initially player 1 wins  $\lambda/2$  (and player 2 loses  $-\lambda/2$ ). The next game has discount factor  $(1 - \lambda)$ ;  $\Gamma^1$  is played with probability  $1/3$  and  $\Gamma^2$  with probability  $2/3$ . Let  $\mathfrak{P}^1 = \mathfrak{P}^1(\alpha, \beta)$  and  $\mathfrak{P}^2 = \mathfrak{P}^2(\alpha, \beta)$  be the discounted pay-offs of player 1, playing strategy  $(\alpha, \beta)$ , starting from  $\Gamma^1$  and  $\Gamma^2$ , respectively. The recursive structure of  $\Gamma[\lambda]$  gives rise to the following equations:

$$\mathfrak{P}^1 = \lambda \frac{1}{2} + (1 - \lambda) \left( \frac{1}{3} \mathfrak{P}^1 + \frac{2}{3} \mathfrak{P}^2 \right), \quad \mathfrak{P}^2 = \lambda 3 + (1 - \lambda) \left( \frac{1}{4} \mathfrak{P}^1 + \frac{3}{4} \mathfrak{P}^2 \right)$$

and so  $\mathfrak{P}^1(\alpha, \beta) = (51 - 39\lambda)/2(11 + \lambda)$  and  $\mathfrak{P}^2(\alpha, \beta) = (51 + 21\lambda)/2(11 + \lambda)$   $\square$

Shapley showed that any stochastic game  $\Gamma = \langle \Gamma^1, \dots, \Gamma^\ell \rangle$  has optimal strategies and a unique value vector  $\mathbf{v} = (v^1, \dots, v^\ell)$ . Given the stochastic game  $\Gamma$ , a numerical vector  $w = (w^1, \dots, w^\ell)$  and a discount factor  $\lambda$ , he defined the zero sum games  $\Gamma^l[\lambda, w] = \langle A_1^l, A_2^l, u^l[\lambda, w] \rangle$  where  $u^l[\lambda, w](i, j) = \lambda u^l(i, j) + (1 - \lambda) \sum_{k=1}^\ell \text{lotteries}_k^l(i, j) w^k$ , and he proved the following theorem that characterizes  $\mathbf{v}$  as a fix point.

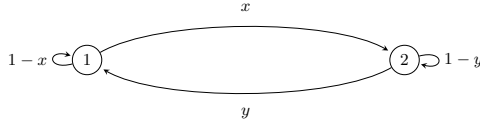
**Theorem 1 ([8]).** *Let  $\Gamma = \langle \Gamma^1, \dots, \Gamma^\ell \rangle$  be a stochastic game and let  $0 < \lambda < 1$ . Then  $\mathbf{v}(\Gamma[\lambda]) = v = (v^1, \dots, v^\ell)$  where  $v$  is the unique solution of the system  $v^l = \nu(\Gamma^l[\lambda, v])$ ,  $l = 1, \dots, \ell$ . One optimal stationary strategy for  $\Gamma[\lambda]$  consists of playing an optimal strategy at node  $l$  for the one shot game  $\Gamma^l[\lambda, v]$ .*

*Example 5.* Let  $w = (w^1, w^2)$ . Then the zero-sum auxiliary games  $\Gamma^1[\lambda, w]$  and  $\Gamma^2[\lambda, w]$  below are derived for the stochastic games in Examples 2 and 4:

$$\begin{array}{c} \begin{array}{cc} & L & R \\ T & \frac{\lambda}{2} + (1 - \lambda) \left( \frac{1}{3} w^1 + \frac{2}{3} w^2 \right) & 1\lambda + (1 - \lambda) \left( \frac{2}{3} w^1 + \frac{1}{3} w^2 \right) \\ B & 1\lambda + (1 - \lambda) \left( \frac{2}{3} w^1 + \frac{1}{3} w^2 \right) & \frac{1}{2} \lambda + (1 - \lambda) \left( \frac{1}{3} w^1 + \frac{2}{3} w^2 \right) \end{array} \\ \Gamma^1[\lambda, w] \\ \begin{array}{cc} & L & R \\ T & 3\lambda + (1 - \lambda) \left( \frac{1}{4} w^1 + \frac{3}{4} w^2 \right) & \frac{3}{2} \lambda + (1 - \lambda) \left( \frac{3}{4} w^1 + \frac{1}{4} w^2 \right) \\ B & \frac{3}{2} \lambda + (1 - \lambda) \left( \frac{3}{4} w^1 + \frac{1}{4} w^2 \right) & 3\lambda + (1 - \lambda) \left( \frac{1}{4} w^1 + \frac{3}{4} w^2 \right) \end{array} \\ \Gamma^2[\lambda, w] \end{array}$$

From Theorem 1 we have,  $\mathbf{v}(\Gamma[\lambda]) = v = (v^1, v^2)$  where  $v^1 = \nu(\Gamma^1[\lambda, v])$  and  $v^2 = \nu(\Gamma^2[\lambda, v])$ . Using the mixed equilibria in Example 1 we have  $v^1 = \frac{3}{4} \lambda + \frac{1}{2} (v^1 + v^2) (1 - \lambda)$  and  $v^2 = \frac{9}{4} \lambda + \frac{1}{2} (v^1 + v^2) (1 - \lambda)$ . Thus,  $\mathbf{v} = (v^1, v^2) = \left( \frac{1}{2} (3 - \frac{3}{2} \lambda), \frac{1}{2} (3 + \frac{3}{2} \lambda) \right)$   $\square$

Oblivious-uncertainty provides a model of the future that is strategy-independent:



**Fig. 1.** A two state regular Markov chain.

**Definition 3.** A stochastic game  $\Gamma = \langle \Gamma^1, \dots, \Gamma^\ell \rangle$  is strategy oblivious if each state  $1 \leq l \leq \ell$ ,  $\text{lotteries}^l$  is a constant function (the same value for all  $(i, j) \in A_1^l \times A_2^l$ ).

Given an oblivious game, let  $\text{vgame}$  be the vector containing the values of the zero-sum state games. Writing  $\text{lotteries}^l = l^l$ , the lotteries corresponding to the different states are given in the following LOTTERIES matrix:

$$\text{vgame} = \begin{pmatrix} \nu(\text{game}^1) \\ \vdots \\ \nu(\text{game}^\ell) \end{pmatrix}, \quad \text{LOTTERIES} = \begin{pmatrix} l^1 \\ \vdots \\ l^\ell \end{pmatrix} = \begin{pmatrix} l_1^1 & \dots & l_\ell^1 \\ \vdots & & \vdots \\ l_1^\ell & \dots & l_\ell^\ell \end{pmatrix}$$

Here LOTTERIES is a stochastic matrix (each row sums to 1) and  $l_k^l$  denotes the probability of moving from state  $l$  to state  $k$ . Recall that a stationary distribution  $\mathbf{p} = (p_1, \dots, p_\ell)$  is a distribution satisfying  $\mathbf{p} \cdot \text{LOTTERIES} = \mathbf{p}$ .

**Theorem 2.** Let  $\Gamma = \langle \Gamma^1, \dots, \Gamma^\ell \rangle$  be a discounted strategy oblivious stochastic game,  $0 < \lambda < 1$ . Then:

1. The value vector  $\mathbf{v} = \mathbf{v}(\Gamma[\lambda])$  satisfies  $\mathbf{v} = \lambda \text{vgame} + (1 - \lambda) \text{LOTTERIES} \cdot \mathbf{v}$
2. The value vector  $\mathbf{v}$  is  $\mathbf{v} = \lambda (I - (1 - \lambda) \text{LOTTERIES})^{-1} \text{vgame}$ .
3. If  $\mathbf{p}$  is a stationary then  $\mathbf{p} \cdot \mathbf{v} = \mathbf{p} \cdot \text{vgame}$ .

In the general case, the non-linearity of the equations in Theorem 1 defining  $\mathbf{v}(\Gamma[\lambda])$  makes it difficult to compute the exact value of a discounted stochastic game. In fact, a stochastic game defined on rational data can have an irrational value vector [11]. In the oblivious case the value vector  $\mathbf{v}$  can be expressed as a linear system over values of local state games (Theorem 2). Thus, as the computation of  $\text{vgame}$  can be reduced to linear programming, assuming strategy obliviousness and rational data the result vector  $\mathbf{v}$  is kept within the rationals.

*Example 6.* Let  $\Gamma = \langle \Gamma^1, \Gamma^2 \rangle$  be a strategy oblivious stochastic game. The matrix LOTTERIES corresponds to a 2 states Markov chain (see Fig. 1). Let

$$\text{LOTTERIES} = \begin{pmatrix} l_1^1 & l_2^1 \\ l_1^2 & l_2^2 \end{pmatrix} = \begin{pmatrix} (1-x) & x \\ y & (1-y) \end{pmatrix}, \quad \mathbf{p} = (p_1, p_2) = \left( \frac{y}{x+y}, \frac{x}{x+y} \right)$$

where  $0 < x, y < 1$  and  $\mathbf{p}$  is a stationary distribution. Suppose that  $\nu(\text{game}^1) = a$  and  $\nu(\text{game}^2) = b$ . The linear system derived from Theorem 2 is:

$$\begin{pmatrix} v^1 \\ v^2 \end{pmatrix} = \lambda \begin{pmatrix} a \\ b \end{pmatrix} + (1 - \lambda) \begin{pmatrix} (1-x) & x \\ y & (1-y) \end{pmatrix} \begin{pmatrix} v^1 \\ v^2 \end{pmatrix}$$

## 4 An Assessment Model for Periodic Orchestrations

In this section an assessment model for orchestrations in stable environments [2] is reviewed and extended to encompass periodic orchestrations in evolving environments.

**Uncertainty profiles and  $\alpha/\delta$ -games.** Let  $E$  be a non-recursive orchestration which, when called, publishes a finite set of results and terminates. The environment for  $E$  is assumed to be uncertain but stable. Let  $\alpha_+(E)$  be the set of sites called by  $E$  (excluding 0). Let  $\#s$  denote the cardinality of a set of sites  $s$ .

The assessment of  $E$  under stress is undertaken by specifying those services which have the potential to be affected by stress. An uncertainty profile  $\mathcal{U}$  models the *a priori* perception of orchestration behaviour under stress [2], providing a model that lies between over-optimism and over-pessimism. A profile  $\mathcal{U}$ , fixes two subsets of  $\alpha_+(E)$ ,  $\mathcal{A}$  and  $\mathcal{D}$ , together with the number of service failures that can be expected to occur within both  $\mathcal{A}$  and  $\mathcal{D}$ . The last component of  $\mathcal{U}$  is a utility  $u$  function which measures resilience under a given type of stress. Behaviour is analyzed by assuming that service failures in  $\mathcal{A}$  (angelic services) are selected to *cause the least amount of damage* whereas service failures in  $\mathcal{D}$  (daemonic services) are selected to *maximise damage to the application*. The assessment of  $E$  goes through a zero-sum game  $\Gamma(\mathcal{U})$ , the  $\alpha/\delta$ -game, providing an analysis of the competitive scenario [2]. Formally:

**Definition 4 (uncertainty profile  $\mathcal{U}$  and its associated  $\alpha/\delta$ -game [2]).**

- An uncertainty profile for an orchestration  $E$  is a tuple  $\mathcal{U} = \langle E, \mathcal{A}, \mathcal{D}, b_{\mathcal{A}}, b_{\mathcal{D}}, u \rangle$  where  $\mathcal{A} \cup \mathcal{D} \subseteq \alpha_+(E)$ ,  $b_{\mathcal{A}} \leq \#\mathcal{A}$ ,  $b_{\mathcal{D}} \leq \#\mathcal{D}$  and  $u(a, d) \geq 0$  is a utility function defined for all  $a \subseteq \alpha_+(E)$ ,  $d \subseteq \alpha_+(E)$ .
- $\mathcal{U} = \langle E, \mathcal{A}, \mathcal{D}, b_{\mathcal{A}}, b_{\mathcal{D}}, u \rangle$  has an associated zero-sum angel-daemon game  $\Gamma(\mathcal{U}) = \langle A_{\alpha}, A_{\delta}, u \rangle$  with two players,  $\alpha$  (angel) and  $\delta$  (daemon). Player  $\alpha$  selects a set with size  $b_{\mathcal{A}}$  from  $\mathcal{A}$ :  $A_{\alpha} = \{a \subseteq \mathcal{A} \mid \#a = b_{\mathcal{A}}\}$ . Player  $\delta$  selects a set with size  $b_{\mathcal{D}}$  from  $\mathcal{D}$ :  $A_{\delta} = \{d \subseteq \mathcal{D} \mid \#d = b_{\mathcal{D}}\}$ . Services in  $\alpha_+(E) \setminus (a \cup d)$  remain reliable.
- The assessment  $\nu(\mathcal{U})$  of an uncertainty profile  $\mathcal{U}$  is defined to be the value of its associated angel-daemon  $\nu(\Gamma(\mathcal{U}))$ .

Utility  $u(a, d)$  measures the degree of resilience of  $E$  when  $\alpha$  "selects" services  $a$  and  $\delta$  "selects" services  $d$ . Different utilities can be used to define different resilience measures. Three utilities with different weightings are shown:

$$u^o(a, d) = \frac{1}{2} \text{out}(\text{fail}_{a \cap d}(E)), u^r(a, d) = \frac{3}{2} \text{out}(\text{fail}_{d \setminus a}(E)), u^w(a, d) = \text{out}(\text{fail}_{a \cup d}(E))$$

Here the function  $\text{out}(\text{fail}_f(E))$  returns the number of outputs published by  $E$  when services in the set  $f$  fail. In the *overloaded* environment,  $u^o$ , services selected by both  $\alpha$  and  $\delta$  fail. In the *robust* environment,  $u^r$ , the angel has the capability to prevent its selected service from failing. In a failures-prone *weak* environment,  $u^w$ , neither  $\alpha$  or  $\delta$  can avoid failures.

*Example 7.*  $BigTwo = (G \mid A)$  can operate in  $\mathcal{U}^o = \langle BigTwo, \{G, A\}, \{G, A\}, 1, 1, u^o \rangle$  describing an *overloaded* environment or in  $\mathcal{U}^r = \langle BigTwo, \{G, A\}, \{G, A\}, 1, 1, u^r \rangle$  giving a *robust* environment or in  $\mathcal{U}^w = \langle BigTwo, \{G, A\}, \{G, A\}, 1, 1, u^w \rangle$  giving a *weak* one. The games  $\Gamma(\mathcal{U}^o)$ ,  $\Gamma(\mathcal{U}^r)$  and  $\Gamma(\mathcal{U}^w)$  are

|     |     |                         |  |  |
|-----|-----|-------------------------|--|--|
|     |     | $\delta$                |  |  |
|     |     | {G} {A}                 |  |  |
| {G} | 1/2 | 1                       |  |  |
| {A} | 1   | 1/2                     |  |  |
|     |     | $\Gamma(\mathcal{U}^o)$ |  |  |

|     |     |                         |  |  |
|-----|-----|-------------------------|--|--|
|     |     | $\delta$                |  |  |
|     |     | {G} {A}                 |  |  |
| {G} | 3   | 3/2                     |  |  |
| {A} | 3/2 | 3                       |  |  |
|     |     | $\Gamma(\mathcal{U}^r)$ |  |  |

|     |   |                         |  |  |
|-----|---|-------------------------|--|--|
|     |   | $\delta$                |  |  |
|     |   | {G} {A}                 |  |  |
| {G} | 1 | 0                       |  |  |
| {A} | 0 | 1                       |  |  |
|     |   | $\Gamma(\mathcal{U}^w)$ |  |  |

Assessments are  $\nu(\mathcal{U}^o) = 3/4$  and  $\nu(\mathcal{U}^r) = 9/4$  (Example 1) and  $\nu(\mathcal{U}^w) = 1/2$ .  $\square$

**Uncertainty profiles for periodic orchestrations.** The assessment of the periodic orchestration is modelled using stochastic games. Possible execution environments of  $E$  are defined by uncertainty profiles.

**Definition 5.** Let  $D = (E \mid (Rtimer(\tau) \gg D))$ . An uncertainty profile for  $E$  is a tuple  $\mathfrak{U} = \langle \mathcal{U}^1, \dots, \mathcal{U}^\ell, \text{lotteries}^1, \dots, \text{lotteries}^\ell \rangle$ , where, for each  $1 \leq i \leq \ell$ ,  $\mathcal{U}^i$  is an uncertainty profile over  $E$  and  $\text{lotteries}^i \in \Delta(\{1, \dots, \ell\})$  is collection of associated lotteries, one lottery for each strategy profile in the game  $\Gamma(\mathcal{U}^i)$ . Profile  $\mathfrak{U}$  induces an associated stochastic  $\alpha/\delta$  game  $\Gamma(\mathfrak{U}) = \langle \Gamma^1, \dots, \Gamma^\ell \rangle$  where, for  $1 \leq i \leq \ell$ ,  $\Gamma^i = \langle \Gamma(\mathcal{U}^i), \text{lotteries}^i \rangle$ . Let  $0 < \lambda < 1$  be a discount value, if  $\mathbf{v}(\Gamma(\mathfrak{U})[\lambda]) = (v^1, \dots, v^\ell)$  then the assessment of  $D$  under  $\mathfrak{U}$  is defined as  $\nu(\mathfrak{U}) = v^1$ .

*Example 8.* The orchestration  $BigTwo^* = (BigTwo \mid (Rtimer(\text{day}) \gg BigTwo^*))$  is assessed by the uncertainty profile  $\mathfrak{U} = \langle \mathcal{U}^o, \mathcal{U}^r, \text{lotteries}^o, \text{lotteries}^r \rangle$ . Here tuples  $\langle \Gamma(\mathcal{U}^o), \text{lotteries}^o \rangle$  and  $\langle \Gamma(\mathcal{U}^r), \text{lotteries}^r \rangle$  correspond respectively to  $\Gamma^1$  and  $\Gamma^2$  components in Example 2. The associated stochastic game has been analyzed in Example 5 where it is shown that  $\mathbf{v}(\Gamma(\mathfrak{U})) = (\frac{1}{2}(3 - \frac{3}{2}\lambda), \frac{1}{2}(3 + \frac{3}{2}\lambda))$ . Therefore the assessment of  $BigTwo^*$  is  $\frac{1}{2}(3 - \frac{3}{2}\lambda)$ .  $\square$

## 5 Example: The Gabrmn System

Gabrmn is an IT system for managing clinical data generated from magnetic resonance spectra [2]. It comprises a number of sub-systems. Clinical data is stored in a sub-system *Databases*. Clinical applications, including *IDL*, are stored on server, *Apps*. A master server, *Proxy*, controls system behaviour. Email is a key service provided by servers *Mail* and *Mirror*; the sub-system  $1(x) < x < (Mail \mid Mirror)$  has built-in redundancy. Service *Backup* allows system recovery to take place. Gabrmn is modelled in Orc as;

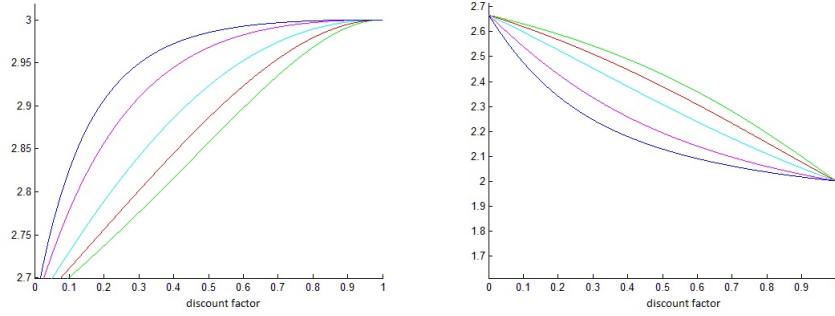
$$IT\_System = Proxy \gg ((1(x) < x < (Mail \mid Mirror)) \mid Apps \mid Backup \mid Databases)$$

This expression is a stylized formalization developed after extensive discussion with the Gabrmn system manager (A. García) - see <http://gabrmn.uab.es/>. The number of outputs ( $u^w = \text{out}(\text{fail}_{a \cup d})$ ) published by *IT\_System* provides a measure of its “well-being” (maximum value 4). The long term behaviour of *IT\_System* in stressed environments is modelled by the following stochastic game:

**States.** Taking  $u^w = \text{out}(\text{fail}_{a \cup d})$ , three different environments for Gabrmn are:

$$\begin{aligned} \mathcal{U}^1 &= \langle IT\_System, \{Backup, Proxy, Mirror\}, \{Apps, Databases, Mail\}, 1, 1, u^w \rangle \\ \mathcal{U}^2 &= \langle IT\_System, \{Apps, Mail, Mirror\}, \{Databases, Mail, Mirror\}, 1, 1, u^w \rangle \\ \mathcal{U}^3 &= \langle IT\_System, \{Apps, Mail, Mirror\}, \{Databases, Mail, Mirror\}, 2, 1, u^w \rangle \end{aligned}$$





**Fig. 2.** Assessments  $v^1$  (left) and  $v^3$  (right) of  $\mathfrak{U}$  with  $x = y = z \in \{1, 0.75, 0.5, 0.25, 0.15\}$ . In both figures the value 1 corresponds to the rightmost curve and 0.15 the the leftmost.

Here a service fails if it is selected by either  $a$  or  $d$ . Profiles  $\mathcal{U}^1, \mathcal{U}^2, \mathcal{U}^3$  induce games with valuations  $\nu(\mathcal{U}^1) = \nu(\mathcal{U}^2) = 3$  and  $\nu(\mathcal{U}^3) = 2$ .

**Evolution of the environment.** The parameterised uncertainty profile  $\mathfrak{U}(x, y, z)$  models system evolution through environments  $\langle \mathcal{U}^1, \mathcal{U}^2, \mathcal{U}^3 \rangle$  using an oblivious approach where

$$\text{vgame} = \begin{pmatrix} \nu(\mathcal{U}^1) \\ \nu(\mathcal{U}^2) \\ \nu(\mathcal{U}^3) \end{pmatrix} = \begin{pmatrix} 3 \\ 3 \\ 2 \end{pmatrix}, \quad \text{LOTTERIES} = \begin{pmatrix} 1-x & x & 0 \\ 0 & 1-y & y \\ z & 0 & 1-z \end{pmatrix}$$

**LOTTERIES** models a perturbed round trip  $1 \rightarrow 2 \rightarrow 3 \rightarrow 1$  with a probability to keep into the current state. The stochastic game  $\Gamma(\mathfrak{U})$  where  $\mathfrak{U} = \langle \mathcal{U}^1, \mathcal{U}^2, \mathcal{U}^3, \text{LOTTERIES} \rangle$  satisfies:

$$\mathbf{v} = \lambda \begin{pmatrix} 1 - (1-\lambda)(1-x) & -(1-\lambda)x & 0 \\ 0 & 1 - (1-\lambda)(1-y) & -(1-\lambda)y \\ -(1-\lambda)z & 0 & 1 - (1-\lambda)(1-z) \end{pmatrix}^{-1} \text{vgame}$$

This equation has been solved, using the Python **SymPy** library, to find the value vector  $\mathbf{v} = (v^1, v^2, v^3)$  for certain values of  $x, y$  and  $z$  and discount factors – see Fig. 2. Setting  $x = 1, y = 1$  and  $z = 1$  gives an environment with a deterministic full round trip (next is to  $1 \rightarrow 2 \rightarrow 3 \rightarrow 1$ ). When  $x = y = z$  the assessment  $v^1$  increases monotonically as the probability of remaining in the first state increases (i.e.  $x$  decreases). As  $\lambda \rightarrow 1$  the assessment  $v^1$  in the discounted model is weighted towards today's performance ( $\nu(\mathcal{U}^1) = 3$  – see Fig. 5 left). Fig. 5 right shows the assessment  $v^3$ , which corresponds to the game starting in the least reliable environment,  $\mathcal{U}^3$ : decreasing the probability of remaining in this state monotonically improves system performance. Similar results have been obtained for asymmetric case:  $y = z$  and  $x = y/2$ .

## 6 Conclusions

The angel-daemon approach has been extended to assess periodic orchestrations in *evolving* environments. We have considered the subfamily of strategy oblivious stochas-

tic games in which transition probabilities are independent of selected strategies. It has been shown that such games have well-defined valuations. The proposed framework has been used to assess the evolution of the Gabrnm IT system. We are in the process of identifying larger systems that can be cast naturally in our approach.

Our approach may go some way to answering a fundamental question of application developers: how can the resilience of service-based systems be assessed when some sub-components are subject to unpredictable forms of stress (e.g. contention for resources on a hypervisor)? Typically practitioners use ad hoc techniques, guided by their technical experience, to develop robust systems of micro-services. It is unclear how the uncertainty associated with a cloud (and services deployed therein) can be modelled realistically by using probabilistic techniques (because sudden surges in demand can occur, or the types of resource available may change in an unpredictable way). Here we provide a different approach which (partially) removes probabilities from the analysis of a natural strategic situation. Our approach provides a formal method for the analysis of an increasingly important class of architectures (until now analysed by trial-and-error techniques). We are working towards extending the approach by adding latency to the set of components that can be influenced by mixed-effect environmental influences.

The work reported here is complementary to that of [1, 7] where the monotonicity of the QoS of web services is considered. One other possible line of future research is to extend the monotonicity properties of the assessments of non-recursive orchestrations [2] to periodic orchestrations.

## References

1. A. Benveniste, C. Jard, A. Kattepur, S. Rosario, and J. A. Thywissen, QoS-aware management of monotonic service orchestrations. *Formal Methods in System Design*, 44(1):1–43, 2014.
2. J. Gabarro, M. Serna and A. Stewart. Analysing web-orchestrations under stress using Uncertainty profiles. *The Computer Journal*, 57(11):1591–1615, 2014.
3. D. Levharit and L. Mirman. The great fish war: an example using a dynamic cournot-nash solution. *The Bell Journal of Economics*, 11:322–334, 1980.
4. J. Misra. A programming model for the orchestration of web services. 2nd Int. Conf. on Software Engineering and Formal Methods, SEFM 2004, pages 2–11. IEEE, 2004.
5. J. Misra and W. Cook. Computation orchestration: A basis for wide-area computing. *Software and Systems Modeling*, 6(1):83–110, 2007.
6. G. Owen. *Game Theory, Third Edition*. Academic Press, Sant Diego, 2001.
7. S. Rosario, A. Benveniste, and C. Jard, Flexible Probabilistic QoS Management of Orchestrations. *Int. J. Web Service Res*, 7(2):21–42, 2010.
8. L. Shapley. Stochastic games. In *PNAS*, pages 1095–1100, 1953.
9. S. Sorin. New approaches and recent advances in two-person zero-sum repeated games. *Annals of the Int. Soc. of Dynamic Games*, 7 (Advances in Dynamic Games):67–93, 2005.
10. A. Stewart, J. Gabarro, and A. Keenan. Reasoning about orchestrations of web services using partial correctness. *Formal Aspects of Computing*, 25:833–846, 2013.
11. O. J. Vrieze. Stochastic games, practical motivation and the orderfield property for special cases. In *Stochastic Games and Applications*, 570 of *NATO Science*, 215–225, 2003.
12. I. Wehrman, D. Kitchin, W. Cook, and J. Misra. A timed semantics of Orc. *Theor. Comput. Sci.*, 402(2-3):234–248, 2008.