

# Probabilistic Resilience in Hidden Markov Models

**Jacopo Panerati and Giovanni Beltrame**

Polytechnique Montreal, 2900 boul. Édouard-Montpetit, Montréal H3T 1J4, Québec, Canada

E-mail: {jacopo.panerati, giovanni.beltrame}@polymtl.ca

**Nicolas Schwind**

Transdisciplinary Research Integration Center, 10-3 Midori-cho, Tachikawa, Tokyo, Japan

E-mail: schwind@nii.ac.jp

**Stefan Zeltner**

Bloomberg L.P., 731 Lexington Ave, New York, NY 10022, USA

E-mail: zeltner@usc.edu

**Katsumi Inoue**

National Institute of Informatics, 2-1-2 Hitotsubashi, Chiyoda, Tokyo 101-0003, Japan

E-mail: inoue@nii.ac.jp

**Abstract.** Originally defined in the context of ecological systems and environmental sciences, resilience has grown to be a property of major interest for the design and analysis of many other complex systems: resilient networks and robotics systems offer the desirable capability of absorbing disruption and transforming in response to external shocks, while still providing the services they were designed for. Starting from an existing formalization of resilience for constraint-based systems, we develop a probabilistic framework based on hidden Markov models. In doing so, we introduce two new important features: stochastic evolution and partial observability. Using our framework, we formalize a methodology for the evaluation of probabilities associated with generic properties, we describe an efficient algorithm for the computation of its essential inference step, and show that its complexity is comparable to other state-of-the-art inference algorithms.

## 1. Introduction and Related Work

Originally defined in the context of ecological systems, resilience is now a property of major interest for the analysis of many complex systems. Its definition is not always unanimous, but researchers agree that resilience is a characteristic property of those ecosystems that, through their history, are able to absorb extreme spikes and, although transformed, survive. The insect populations of North-eastern American forests [1] are notable examples of such sustainable systems. In the coming years, “resilient by design” networks and robotics system will possess the ability to absorb disruption and to transform in response to external shocks, while still providing their services. Resilience will also be implemented in welfare measures—such as flood prevention plans and healthcare systems—helping making human society a more sustainable system too.



So far, the effort of the artificial intelligence community has been focused on narrowing down the concept of resilience through its formalization, e.g., in constraint-based and non-deterministic dynamic systems [2]. This approach is extremely general and it can be adapted to describe the behaviour of a plethora of real-world systems but it comes at the cost of a limited predictive power. The transition models in non-deterministic dynamic systems resemble those of Markov decision processes but, lacking probability values associated to transitions, they do not tell which future worlds are the most likely.

By exploiting hidden Markov models, we preserve much of the existing research and include a fundamental aspect that was not investigated in earlier works: the unpredictability of the real world. This is essential for any modelling that aspires to be consistent with the idea of a “random world” proposed by Holling [1] and conditional probability distributions can be seen as the stochastic extension of the non-deterministic transition functions. We also enrich the existing discussion about resilience with another important layer of complexity, that is, partial observability. The goal of our work is to be able to efficiently answer queries such as: “what is the probability that Canada’s inflation rate will not drop below 1% in the next five years?” or “with 95% confidence, what is the minimum size of a certain wild animal population, only rarely observed?”.

In a seminal paper from 1973, Holling introduced the concept of “resilience of ecological systems”. Resilient systems are not those systems that simply react to imbalances by quickly returning to equilibria. Instead, when perturbed, they are able to find new sustainable configurations. It is worth noting that Holling defines resilience in the context of what he calls “the random world”: an environment that is intrinsically stochastic. Walker *et al.* [3] define resilience as “the capacity of a system to absorb disturbance and reorganize while undergoing change so as to still retain essentially the same function, structure, identity, and feedbacks”.

The first attempt to formally define the concept of resilience exploiting the tools of artificial intelligence was provided by Överén, Willsky, and Antsaklis [4] and successively elaborated by Baral *et al.* [5] and Schwind *et al.* [2]. The SR-model, proposed in [2], is the starting point of our work and it is described in more detail in the background section. The most relevant difference between these works and our methodology is the introduction of the ideas of probability theory in the latter. The probabilistic framework on which we develop our analysis is that of hidden Markov models. Their traditional applications are in signal and natural language processing, however, their use to describe more complex dynamic systems has also proven to be fruitful [6].

The two most common types of inference tasks for probabilistic graphical models are marginal and maximum a posteriori estimation. For these, HMMs provide efficient algorithms that have convenient linear time-complexity [7]. In this work, we are interested in the probability of properties lasting over time, therefore, we cannot simply apply the state-of-the-art algorithms for marginals or maximum a posteriori. Nonetheless, we show that the queries we need can be implemented by an algorithm in the same complexity class (i.e. requiring the same amount of computational resources).

## 2. Background

Before delving into the proposed methodology, we briefly review the theoretical background on the two main aspects of this work: 1) the formal definition of resilience in sustainable systems; 2) and hidden Markov models.

The SR-model is a theoretical framework proposed by Schwind *et al.* [2] combining elements of constraint-based systems and non-deterministic dynamic systems. It provides a formal definition of resilience as a unifying property combining different aspects of three simpler properties: resistance, functionality, and recoverability.

*2.0.1. Kinematics and dynamics* The SR-model provides two separate formal descriptions for the kinematics and the dynamics of a system: the first consists of unbounded sequences of state-couples called “state trajectories” or SSTs:

$$SST = (CBS_1, \gamma_1), (CBS_2, \gamma_2), \dots \quad (1)$$

The subscript index skims through discrete time steps. Each  $CBS_i$  represents a constraint-based system composed of a set of variables  $\mathbf{X}_i$  and a cost function  $c_i$ :

$$CBS_i = (\mathbf{X}_i = \{X_i^1, X_i^2, \dots\}, c_i : \mathcal{D}(\mathbf{X}_i) \rightarrow \mathbb{R}^+). \quad (2)$$

Finally,  $\gamma_i \in \mathcal{D}(\mathbf{X}_i)$  represents a complete assignment of the variables in  $\mathbf{X}_i$ :

$$\gamma_i \in \mathbb{R}^{|\mathbf{X}_i|} \quad (3)$$

As a consequence, each SST corresponds unambiguously to a sequence of costs obtained by plugging-in each  $\gamma_i$  into the corresponding cost function  $c_i$ :

$$c_1(\gamma_1), c_2(\gamma_2), \dots \quad (4)$$

The environment dynamics are described through a non-deterministic dynamic system DS:

$$DS = \langle \mathbf{CBS}, \mathbf{A}, t : \mathbf{CBS} \times \mathbf{A} \rightarrow \mathcal{P}(\mathbf{CBS}) \rangle \quad (5)$$

where  $\mathbf{CBS}$  represents the set of all possible constraint-based systems  $CBS_i$ ,  $\mathbf{A}$  is the set of actions available at each time step, and  $t$  is a non-deterministic transition function that, given the current constraint-based system and an action, returns the set of possible constraint-based systems at the next time step.

The kinematic description (SSTs and sequences of costs) is essential for understanding the resilient properties and it is preserved in our proposed methodology. In this work, however, the non-deterministic description of the dynamics is discarded for a probabilistic approach.

*2.0.2. The resilient properties* In this model, resilience is a boolean property of a state trajectories SST. It can be seen as a unifying property, combining different desirable behaviours of a dynamic system and it arises from three simpler properties of state trajectories: resistance, functionality, and recoverability (see Figure 1).

*l-resistance* The resistance property expresses the fact that a trajectory never incurs in a cost that is larger than a fixed threshold. Therefore, this property is parametrized by this maximum acceptable cost.

**Definition 1** Given a state trajectory  $SST = (CBS_1, \gamma_1), (CBS_2, \gamma_2), \dots$  and a positive threshold  $l \in \mathbb{R}^+$ , SST is said to be *l-resistant* if and only if each cost in its corresponding cost sequence is less than or equal to the threshold  $l$ :

$$c_i(\gamma_i) \leq l \quad \forall c_i(\gamma_i) \in (c_1(\gamma_1), c_2(\gamma_2), \dots) \quad (6)$$

For example, any organization (e.g. the army of a sovereign nation) whose budget is assigned periodically (e.g. yearly) by and external entity (e.g. the Department of Defense) requires to satisfy this property.

*f*-functionality The functionality property tells us if the costs of a trajectory are, on average, equal to or below a certain threshold. As in the case of resistance, this threshold parametrizes the property.

**Definition 2** Given a state trajectory  $SST = (CBS_1, \gamma_1), (CBS_2, \gamma_2), \dots$  and a positive threshold  $f \in \mathbb{R}^+$ ,  $SST$  is said to be *f*-functional if and only if the arithmetic average of the costs in its corresponding cost sequence is less than or equal to the threshold  $f$ :

$$\frac{1}{|SST|} \sum_{i=1}^{|SST|} c_i(\gamma_i) \leq f \quad (7)$$

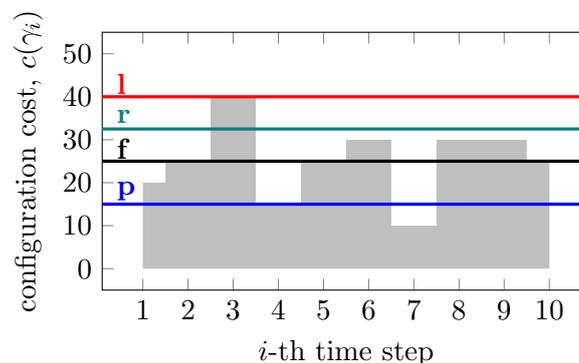
This property is important when we plan operations that span over multiple time steps (e.g a trip lasting several days or weeks) on a fixed budget. From time to time, we might allow ourselves to spend more than the daily allowance: what matters is our total expenditure (or the global average).

$\langle p, q \rangle$ -recoverability The recoverability property concerns those systems in which costs over a certain threshold can be accepted, but only as long as the system is able to return within normal conditions before consuming a fixed, restorable, reserve.

**Definition 3** Given a state trajectory  $SST = (CBS_1, \gamma_1), (CBS_2, \gamma_2), \dots$ , a positive threshold  $p \in \mathbb{R}^+$  and a positive budget  $q \in \mathbb{R}^+$ ,  $SST$  is said to be  $\langle p, q \rangle$ -recoverable if and only if every time the sequence of costs exceeds the threshold, it also returns below (or at) it before that the cumulative offset surpasses the reserve:

$$\forall k \text{ s.t. } c_k(\gamma_k) > p, \exists j > k \text{ s.t. } c_j(\gamma_j) \leq p \wedge \sum_{i=k}^{j-1} (c_i(\gamma_i) - p) \leq q \quad (8)$$

An example of a recoverable system is the human muscle tissue: it can perform at maximum intensity for a short time consuming a molecule called *adenosine triphosphate*, or ATP [8], which is available in limited quantities in our body. To recover, the muscle needs to decrease the intensity of the effort, and allow other metabolic pathways to replenish the initial ATP storage.



**Figure 1.** Example of a cost trajectory that is 40-resistant, 25-functional,  $\langle 15, 40 \rangle$ -recoverable

Hidden Markov models (HMMs) can be seen a sub-family of both dynamic Bayesian networks (DBNs) and state-observation models [7]. HMMs have a single discrete state variable  $S$  and a single discrete observation variable  $O$ . A hidden Markov model is fully specified by the

probability distribution of  $S$  at time 0, the conditional distribution of  $O$  given  $S$  at the same time step, and the conditional distribution of  $S$  given  $S$  at the previous time step [9].

$$HMM = \langle P(S_0), P(O_t | S_t), P(S_{t+1} | S_t) \rangle \quad (9)$$

HMMs are commonly used for the tasks of signal processing and speech recognition [9] because efficient (i.e. with computational time complexity that is linear with respect to the time horizon of the model) algorithms exist for: 1) the estimation of the probability distribution of  $S$ , also called the “hidden” variable, taking only assignments of  $O$  as input (filtering and smoothing algorithms); and 2) the identification of the most likely sequence of assignments of  $S$ . To formalize resilient properties in a probabilistic context, HMMs offer the probabilistic reasoning of DBNs and the independence assumptions of state-observation models, while minimizing representation size.

### 3. Methodology

In this section, we identify the shortcomings of the traditional HMM framework and suggest how to improve it. On top of this extended model, we provide the definitions of trajectory of states  $TS$  and trajectory of observations  $TO$ , and we show how to compute their probability values.

#### 3.1. Framework Extensions

We can imagine the single discrete state variable  $S$  of a HMM as a way to represent, by enumeration, the state configuration that was encoded by the set of variables  $\mathbf{X}$ , and assignment  $\gamma$ , in the constraint-based systems  $CBSs$ . However, HMMs do not natively provide an important element of the SR-model: a cost function  $c$  to state the expensiveness, for the purpose of resilience, of each assignment. In the following, the extension of the HMM framework with a static cost function  $c$ , defined over the domain  $\Omega$  of its (random) state variable  $S$  and taking positive values in  $\mathbb{R}$ , is called c-HMM (see Figure 2) and formally defined as:

$$\langle P(S_0), P(O_t | S_t), P(S_{t+1} | S_t), c : \Omega(S) \rightarrow \mathbb{R}^+ \rangle \quad (10)$$

A static, i.e. non time-varying, cost function means that our preferences do not change over time.

#### 3.2. Trajectories of States, Observations, and Costs

In the SR-model, the definition of the resilient properties is based on the concepts of SSTs and their corresponding sequences of costs. In c-HMMs, we have similar constructs for states and costs, as well as observations. The main aspect of novelty here is that these concepts are now built on top of random variables [10] and, therefore, they also can be associated to probability distribution functions. Given a c-HMM and a finite time horizon  $T$ , we define its trajectory of states  $TS$  as the sequence of state variables  $S_i \forall i \in \{1, \dots, T\}$ . This can be rewritten as:

$$TS := S_1, S_2, \dots, S_T \quad (11)$$

Because  $S$  is a random variable,  $TS$  is also a random variable that can take as value any possible sequence of assignments of  $S$  of length  $T$ , i.e.  $ts = s_1, s_2, \dots, s_T$ . Therefore, the number of possible assignments of  $TS$  grows exponentially with the time horizon:

$$|\Omega(TS)| = |\Omega(S)|^T \quad (12)$$

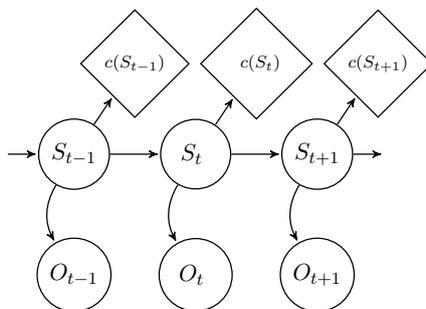
Because the mapping provided by the cost function  $c$  is entirely deterministic, each assignment of  $TS$  is unambiguously associated to a trajectory of costs  $tc = c(s_1), c(s_2), \dots, c(s_T)$  and  $TC$  is also

a random variable with  $|\Omega(TC)| \leq |\Omega(TS)| = |\Omega(S)|^T$ . Considerations similar to those we made for trajectories of states are also valid for trajectories of observations ( $TO := O_1, O_2, \dots, O_T$ ) and their possible assignments ( $to := o_1, o_2, \dots, o_T$ ). If neither the transition model or the sensor model contain probability values of 0, all possible sequences of states can occur and produce any of the sequences of observations. Therefore, the number of possible configurations of a c-HMM is:

$$(|\Omega(S)| \cdot |\Omega(O)|)^T \quad (13)$$

This represents how difficult inference, in general, could be in our model. We also observe that: 1) the state variable  $S$  is called the “hidden” variable because it is never directly observable, meaning that which  $ts$  value is actually taken by  $TS$  is unknown too; while 2) the values taken by the observations variable  $O$  are the pieces of *partial* information that we can always access. Therefore, the kind of probabilistic inference we are interested in is the one dealing with the  $|S|^T$  probability values of the conditional probability distribution of  $TS$  with respect to a given  $to$ :

$$P(TS | to) = P(S_1, S_2, \dots, S_T | o_1, o_2, \dots, o_T) \quad (14)$$



**Figure 2.** The extended c-HMM framework unrolled over three time steps.

### 3.3. Probability of Cost Trajectories and Properties

So far, we have seen that the resilient properties, once their parameters are fixed, can be considered as boolean attributes of sequences of costs associated to SSTs. In the context of the c-HMM framework, we say that an assignment of the trajectory of states  $ts = s_1, s_2, \dots$  enforces the property  $\pi$  if and only if its corresponding trajectory of costs  $tc = c(s_1), c(s_2), \dots$  is satisfies the definition of that property, i.e.  $\pi(c(s_1), c(s_2), \dots) = true$ . Computing the probability distribution of parametric properties  $\pi(k)$ , such as resistance and functionality, with respect to their parameters, can provide valuable insights, as shown in Figure 3.

The probability of property  $P(\pi)$  is equal to the sum of the probabilities of all the distinct trajectories of costs in which  $\pi$  holds:

$$P(\pi) = \sum_{\forall i \in \{i | \pi(tc_i) = true\}} P(tc_i) \quad (15)$$

In turn, the probability of a fixed assignment of the trajectory of costs  $tc$  is equal to the sum of the probabilities of all the distinct trajectories of states that are mapped to it by the cost function  $c$ . To simplify the notation, we will also use  $C(ts)$  to indicate the  $tc = c(s_1), c(s_2), \dots$  resulting from the application of the cost function  $c$  to SST assignment  $ts$ .

$$P(tc) = \sum_{\forall i \in \{i | C(ts_i) = tc\}} P(ts_i) \quad (16)$$

By plugging Equation 16 into Equation 15 we can compute the probability of a property  $\pi$  as a function of the probabilities of distinct assignments of the trajectory of states (Equation 17). We observe that all the possible assignments of  $TS$  are “distinct” by definition, even if many of them could be mapped by  $c$  to identical trajectories of cost.

$$P(\pi) = \sum_{\forall i \in \{i | \pi(tc_i) = true\}} \sum_{\forall j \in \{j | C(ts_j) = tc_i\}} P(ts_j) \quad (17)$$

### 3.4. Conditioning w.r.t. the Trajectory of Observations

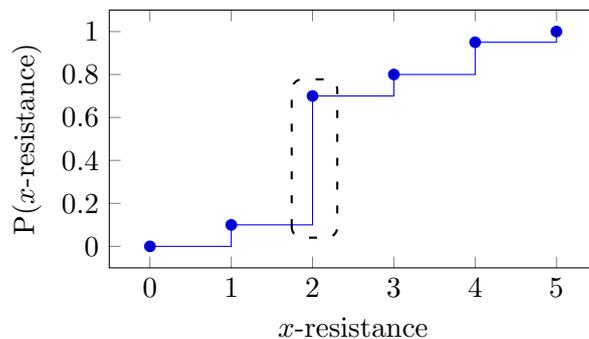
When we are able to observe the system, we become interested in computing the conditional distribution of  $\pi$  with respect to the assignment of the trajectory of observations  $to$ . To do so, we need to update Equation 17 by conditioning on both sides using  $to$ :

$$P(\pi | to) = \sum_{\forall i \in \{i | \pi(tc_i) = true\}} \sum_{\forall j \in \{j | C(ts_j) = tc_i\}} P(ts_j | to) \quad (18)$$

Equation 18, shows that computing the probability of a property  $\pi$ , given an assignment of the trajectory of observations  $to$ , consists of two different subproblems:

- identifying which assignments of the trajectory of states  $TS$  map to assignments of the trajectory of costs  $TC$  such that the property is satisfied;
- computing the conditional probability of these assignments of  $TS$  with respect to the assignment of the trajectory of observations  $to$ .

The problem at point a) strictly depends on the nature of the property we are evaluating and it is discussed further later on. Instead, we now approach the second problem which is more general and can be addressed combining different inference methods for HMMs.



**Figure 3.** Probability distribution of parametric resistance in an example where  $\forall s, c(s) \in [1, \dots, 5]$ . The discontinuity around 2 shows the critical threshold for the property.

#### 3.4.1. Conditional probability of a finite assignment of the state trajectory: $P(s_1, \dots, s_T | o_1, \dots, o_T)$

A highly inefficient way to find this probability value consists of computing the complete joint probability distribution (JPD) of the c-HMM over the time horizon  $T$ —because HMMs are Bayesian networks, their JPD is equal to the chain product of all the conditional probability distributions (CPDs) in their nodes—then, conditioning by the evidence of  $to$ , and finally re-normalizing the entire distribution. This is not a good approach because computing the JPD requires time and space complexity of  $(|S| \cdot |O|)^T$ . A more efficient approach to the computation of the conditional probability of a finite state trajectory assignment  $s_1, \dots, s_T$  with respect to

the observation trajectory assignment  $o_1, \dots, o_T$  can be found using the Bayes' theorem:

$$\begin{aligned} P(s_1, \dots, s_T \mid o_1, \dots, o_T) &= \\ &= \frac{P(o_1, \dots, o_T \mid s_1, \dots, s_T) \cdot P(s_1, \dots, s_T)}{P(o_1, \dots, o_T)} \end{aligned} \quad (19)$$

Equation 19 shows how to decompose the problem into three simpler computations that can be tackled separately.

*Probability of:  $o_1, \dots, o_T \mid s_1, \dots, s_T$*  Given an assignment of the state variables, the conditional probability of a sequence of observations can be computed as the product of the appropriate entries of the sensor model.

$$P(o_1, \dots, o_T \mid s_1, \dots, s_T) = \prod_{i=1}^T P(O_t = o_i \mid S_t = s_i) \quad (20)$$

*Probability of:  $s_1, \dots, s_T$*  The probability of an assignment of the the trajectory of states  $TS$ , notwithstanding the values taken by the observations variables, only depends on the transition model  $P(S_{t+1} \mid S_t)$  and the ground belief  $P(S_0)$ . First, we need to compute the probability of  $S_1$  taking the value of  $s_1$  as  $P(S_1 = s_1) = \sum_{\forall s \in \Omega(S)} P(S_{t+1} = s_1 \mid S_t = s)P(S_0 = s)$ ; then, the probability of the entire trajectory can be computed multiplying the appropriate entries of the transition model:

$$P(s_1, \dots, s_T) = P(S_1 = s_1) \cdot \prod_{i=2}^T P(S_{t+1} = s_i \mid S_t = s_{i-1}) \quad (21)$$

*Probability of:  $o_1, \dots, o_T$*  The computation of this latter factor is the trickiest: it can be performed efficiently by applying a dynamic programming technique called Forward algorithm [11]. This algorithm iteratively computes the quantity  $P(o_1, \dots, o_T, s_T)$ —which is, from now on, re-written as  $F(T, s_T)$ —using the transition and sensor models. The initialization step of the Forward algorithm is:

$$\forall x \in \Omega(S) \quad F(1, x) = P(S_1 = x) \cdot P(O_1 = o_1 \mid S_1 = x) \quad (22)$$

The distribution of  $P(S_1)$  can be computed in the same way we did for Equation 21. Then, the  $t + 1$  iteration step of the Forward algorithm is:

$$\begin{aligned} \forall x \in \Omega(S) \quad F(t+1, x) &= \\ &= \sum_{\forall y \in \Omega(S)} F(t, y) \cdot P(S_{t+1} = x \mid S_t = y) \cdot P(O_t = o_t \mid S_t = x) \end{aligned} \quad (23)$$

Finally, having iterated the algorithm until  $T$ ,  $P(o_1, \dots, o_T)$  is computed as the sum over all possible values of  $S_T$ :

$$P(o_1, \dots, o_T) = \sum_{\forall x \in \Omega(S)} F(T, x) \quad (24)$$

### 3.5. Considerations on Time Complexity

Having how to compute  $P(s_1, \dots, s_T \mid o_1, \dots, o_T)$ , we now examine its efficiency. We assume that querying the sensor and the transition models for any of their elements involves a constant and negligible delay. The computation of the  $P(o_1, \dots, o_T)$  factor is the quickest: starting with an initialization value of 1, we need to multiply it  $T$  times by the correct entry of the sensor model. Therefore, the time complexity of computing  $P(o_1, \dots, o_T)$  is  $O(T)$ . The factor  $P(s_1, \dots, s_T)$  is obtained through  $|\Omega(S)|$  multiplications and  $|\Omega(S)| - 1$  sums, to find  $P(S_1 = s_1)$ , and  $T - 1$  products by entries of the transition model. Its time complexity is equal to  $O(|\Omega(S)| + T)$ . Finally, the Forward algorithm has a run time of  $O(|\Omega(S)|^2 \cdot T)$  [11], plus  $|\Omega(S)| - 1$  additions to compute  $P(o_1, \dots, o_T)$ . Hence, the computation of  $P(o_1, \dots, o_T)$  using the Forward algorithm is the slowest of the three. Indeed, this is also the overall time complexity of  $P(s_1, \dots, s_T \mid o_1, \dots, o_T)$ :

$$\text{time-complexity: } O(|\Omega(S)|^2 \cdot T) \quad (25)$$

This result arises from the consideration that all the three computations of Bayes' theorem factorization in Equation 19 are independent (from a computational point of view, not with regard to probability) and, given sufficient resources, they can be performed in parallel, with the last one strictly dominating the others.

Most importantly, we observe that, if the time horizon is much larger than the number of states (i.e.  $T \gg |\Omega(S)|$ ), this probabilistic inference algorithm has an overall time complexity dominated by  $O(T)$ . This result places our algorithm in the same time complexity class of other well-known inference algorithms for HMMs: the Forward-backward algorithm for the computation of smoothed marginals distributions,  $P(S_i \mid to)$ ; and the Viterbi algorithm for the computation of the most likely sequence of hidden variables,  $\text{argmax}_{TS} P(TS \mid to)$  [7].

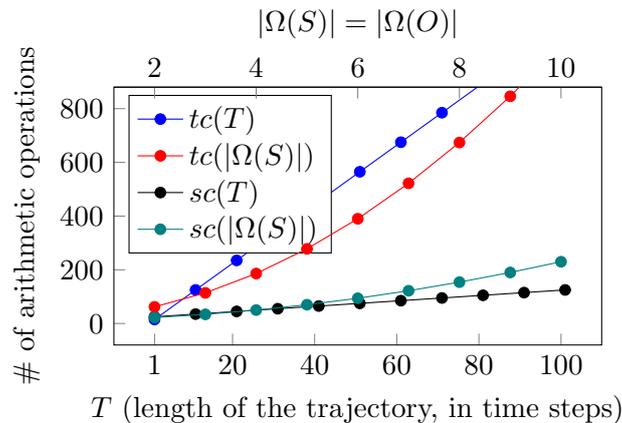
### 3.6. Considerations on Space Complexity

With regard to space complexity, we first have to consider the data structures necessary to represent the c-HMM framework:  $P(S_0)$  has size of  $O(|S|)$ ,  $P(O_t \mid S_t)$  of  $O(|\Omega(S)| \cdot |\Omega(O)|)$ ,  $P(S_{t+1} \mid S_t)$  of  $O(|\Omega(S)|^2)$ , and  $c$  of  $O(|\Omega(S)|)$ . The input of our inference queries consists of two vectors,  $s_1, \dots, s_T$  and  $o_1, \dots, o_T$ , having size of  $O(T)$  each. The computation of  $P(o_1, \dots, o_T)$  only requires to iteratively multiply the result of a previous product and finally store this single floating point value, hence, its space complexity is  $O(1)$ . Similarly, the factor  $P(s_1, \dots, s_T)$  can be computed by repeatedly storing the result of successive additions and multiplications in the same memory cell and it has space complexity of  $O(1)$ . Finally, the execution of the Forward algorithm demands a chunk of memory of  $O(|\Omega(S)|)$ , again dominating the other two factors.

As a results, the performance of the computation of  $P(s_1, \dots, s_T \mid o_1, \dots, o_T)$  can be optimized by using *ad hoc* memory components for its three parts:

$$\text{space-compl. } \begin{cases} 1) \text{ model: } O(|\Omega(S)| \cdot |\Omega(O)| + |S|^2) \\ 2) \text{ input: } O(T) \\ 3) \text{ algorithm: } O(|\Omega(S)|) \end{cases} \quad (26)$$

This also means that the memory requirements of the inference algorithm itself do not depend on the time horizon  $T$ . In most practical cases, in which  $T \gg |\Omega(S)|$ , the memory bottleneck will be represented by the memories dedicated to the storage of the input sequences  $s_1, \dots, s_T$  and  $o_1, \dots, o_T$ . Figure 4 summarizes the time and space complexity relationships of our algorithm and its input.



**Figure 4.** Complexity growth of the inference algorithm with respect to the time horizon  $T$  and the size of the state domain  $|S|$ . In the legend,  $tc$  and  $sc$  are abbreviations for time-complexity and space-complexity, respectively.

#### 4. Conclusions

In this work, we proposed to formalize the resilient properties of resistance, functionality, and recoverability over the timed probabilistic framework of hidden Markov models. To do so, we extended the HMM definition to include a cost function. Then, we defined an inference algorithm able to answer the queries required for probabilistic property checking over this model. The algorithm has been implemented and tested in the MATLAB-compatible scripting language Octave. Finally, we analyzed the space and time complexity of this inference algorithm, as well as the specific complexity of checking generic properties.

#### References

- [1] Holling C S 1973 *Ann. Rev. of Ecology and Systematics* **4** 1–23
- [2] Schwind N, Okimoto T, Inoue K, Chan H, Ribeiro T, Minami K and Maruyama H 2013 Systems resilience: A challenge problem for dynamic constraint-based agent systems *Proceedings of the 2013 International Conference on Autonomous Agents and Multi-agent Systems AAMAS '13* (Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems) pp 785–788 ISBN 978-1-4503-1993-5
- [3] Walker B, Holling C S, Carpenter S R and Kinzig A 2004 *Ecology and Society* **9**(2)
- [4] Özveren C M, Willsky A S and Antsaklis P J 1991 *J. ACM* **38** 729–751 ISSN 0004-5411
- [5] Baral C, Eiter T, Bjrelund M and Nakamura M 2008 *Artificial Intelligence* **172** 1429 – 1469 ISSN 0004-3702
- [6] Rabiner L 1989 *Proceedings of the IEEE* **77** 257–286 ISSN 0018-9219
- [7] Russell S J and Norvig P 2003 *Artificial Intelligence: A Modern Approach* 2nd ed (Pearson Education) ISBN 0137903952
- [8] Gaitanos G C, Williams C, Boobis L H and Brooks S 1993 *Journal of Applied Physiology* **75** 712–719 ISSN 8750-7587
- [9] Koller D and Friedman N 2009 *Probabilistic Graphical Models: Principles and Techniques - Adaptive Computation and Machine Learning* (The MIT Press) ISBN 0262013193, 9780262013192
- [10] Ross S M 2006 *Introduction to Probability Models, Ninth Edition* (Orlando, FL, USA: Academic Press, Inc.) ISBN 0125980620
- [11] Durand D and Hoberman R 2011 Computational genomics and molecular biology, hmm lecture notes Tech. rep. Carnegie Mellon University