

A Dynamic Bayesian Network to represent Discrete Duration Models

Roland Donat^{a,b}, Philippe Leray^c, Laurent Bouillaut^a, Patrice Aknin^a

^a*Laboratoire des Technologies Nouvelles, Institut National de Recherche sur les Transports et leur Sécurité – 2 rue de la Butte Verte - Descartes 2, F-93166 Noisy-le-Grand Cedex*

^b*Laboratoire d'Informatique, Traitement de l'Information et des Systèmes, EA 4108 – Institut National des Sciences Appliquées de Rouen - avenue de l'université, 76801 Saint Etienne du Rouvray*

^c*Équipe COonnaissances et Décision, Laboratoire d'Informatique de Nantes Atlantique, UMR 6241 – Site de l'École Polytechnique de l'Université de Nantes, La Chantrerie - Rue Christian Pauc - BP 50609, 44360 Nantes Cedex 3*

Abstract

Originally devoted to specific applications such as biology, medicine and demography, duration models are now widely used in economy, finance or reliability. Recent works in various fields of application have shown the relevancy of using Bayesian networks to model complex systems, namely stochastic systems with an underlying distribution that does not fit to a well-known parametric form. In this paper, the description of a specific dynamic Bayesian network, referred to as Graphical Duration Model (GDM), is given. A GDM aims to represent a wide range of duration models. Its structure allows especially to fit multi-state systems featuring complex sojourn-time distributions and contextual dependencies. To that end, a duration variable is explicitly introduced to the state transition model which is classically represented by a Markov chain. A recursive algorithm to efficiently perform inference in this model is derived along with its proof of correctness and space and time complexity studies. Finally, this approach is illustrated with an application in survival analysis in which the proposed model is compared with the commonly used Markov chain modelling.

Key words: dynamic Bayesian networks, graphical duration models, discrete duration models, inference, survival analysis

1. Introduction

Discrete duration models are well-fitted to phenomena which can be represented by a positive integer value random variable. The duration model framework provides effective tools when the question is to model and estimate a probability distribution describing the duration between two events (e.g. the lifetime of an infected patient, the time to failure of a physical system, the duration of an unemployment period, the time-interval between a loan acceptance

and a repayment failure, the time-interval between volcanic eruptions ...). Duration models are also relevant to represent transitions between two events (e.g. the transition between two degradation states of a production system, the transition between two phases of a disease, the behavioral change of a poker player after several significant losses ...).

Although originally related to biology, medicine or demography, duration models are intensively used in many fields of application such as economy, finance or reliability [10]. It is important to notice that the approach presented in the sequel can be applied in many domains even if we have chosen to illustrate our purpose with a survival analysis [5].

Recent studies involving the use of Bayesian networks [13, 8] have been proved relevant to represent and analyse complex systems. By complex system we mean stochastic systems with an underlying distribution that does not fit to a well-known parametric form. For instance the authors in [2] show how to model the dependability of an industrial system by means of Bayesian networks. In [14] experiments using dynamic Bayesian networks [12] have been carried out to study the lifetime of a dynamic system represented by a Markov chain. The interest of using these tools stems from (i) the powerful and intuitive graphical modelling capabilities (ii) the generic learning and inference tools allowing respectively to fit the model parameters and to perform stochastic calculations.

In this article, we present a specific dynamic Bayesian network structure, denoted by graphical duration model. This structure is aimed to improve the graphical representation of a Markov chain by introducing a sojourn-time variable depending on the system state. The sojourn-time variable allows to explicitly defined the distribution of the time spent in each system state. Thereby complex stochastic processes with any sojourn-time distributions can be captured by the proposed model. An *ad hoc* inference procedure is provided, along with its proof of correctness, to efficiently compute the probability of any state trajectories through this structure.

To that end, this paper is organised as follows. Section 2 briefly gives some elements of the Bayesian network theory. Section 3 introduces in details the proposed graphical duration model. Section 4 describes the inference algorithm dedicated to this specific structure. Section 5 presents a comparison between the proposed modelling approach and the Markov chains framework applied to survival studies. Finally, some conclusions and perspectives are discussed in section 6.

2. Bayesian networks

Bayesian Networks (BNs) are mathematical tools relying on both the probability theory and the graph theory [13, 8]. BNs allow to qualitatively and quantitatively represent uncertainty. Basically, BNs are used to compactly describe the joint distribution of a collection of random variables $\mathbf{X} = (X_1, \dots, X_D)$ taking their values in the domain $\mathcal{X} = \mathcal{X}_1 \times \dots \times \mathcal{X}_D$. Formally, a BN, denoted by \mathcal{M} , is defined as a pair $(\mathcal{G}, (P(X_d|\mathbf{X})_{\text{pa}_d})_{1 \leq d \leq D})$ where (i) $\mathcal{G} = (\mathbf{X}, \mathcal{E})$ is

a directed acyclic graph in which $\mathbf{X} = (X_1, \dots, X_D)$ is the sequence representing both the graph nodes and the random variables, and \mathcal{E} is the set of edges encoding the conditional dependence relationships among these variables (ii) $P(X_d|\mathbf{X}_{\text{pa}_d})$ is the Conditional Probability Distributions (CPD) of variable X_d given its parents \mathbf{X}_{pa_d} where pa_d denotes the indices sequence of the parents of X_d in \mathcal{G} . These CPDs aim to quantify the "strength" of the dependence relationships between the variables.

The conditional independence relationships associated to the BN graphical structure allow to parsimoniously rewrite the joint distribution of (X_1, \dots, X_D) as follows :

$$P(X_1, \dots, X_D) = \prod_{d=1}^D P(X_d|\mathbf{X}_{\text{pa}_d}). \quad (1)$$

Note that if each random variable X_d takes its values on a discrete and finite set \mathcal{X}_d , the corresponding BN is said to be a discrete and finite BN (DF-BN). In the sequel, the number of possible values for a discrete and finite random variable X over domain \mathcal{X} will be denoted by $N_X = \text{card}(\mathcal{X})$. Moreover in this case, each CPD $P(X_d|\mathbf{X}_{\text{pa}_d})$ can be represented as a matrix with each parent configuration $\mathbf{x}_{\text{pa}_d} \in \mathcal{X}_{\text{pa}_d}$ along the rows and each variable value $x_d \in \mathcal{X}_d$ along the columns.

Besides, both the structure and probabilistic part of a BN can be automatically learnt if some complete or incomplete data, or expert opinions are available [9]. Once the parameters of the CPDs are estimated, the objective is often to perform probabilistic inference. In other words, the question is to compute the distribution of any variables vector $\mathbf{X}_q \subseteq \mathbf{X}$.

One of the most classical inference procedures relies on the use of a junction tree [11]. Nevertheless in our experiments, we use the elimination algorithm [6]. This choice is motivated by the simplicity and the efficiency of the method. Indeed, it has the same interesting computational properties as the junction tree algorithm [7], even though no complex graph theory background is needed to understand and implement it.

Note such modelling is not adapted to time-dependant processes. This drawback can be very problematic. For instance in survival analysis, the question is often to capture how an infection evolves from one particular phase to another over time.

To overcome this limitation, let introduce the Dynamic Bayesian Networks (DBNs) [12]. The DBN formalism is based on the previously defined Bayesian networks with extensions to represent discrete sequential systems. Indeed, DBNs allow to model data which are sequentially generated by some complex mechanisms (e.g. number of mechanical solicitations before the failure of a device, bio-sequences ...).

More precisely, a DBN defines the probability distribution of a collection of random variables $(\mathbf{X}_t)_{t \geq 1} = (X_{1,t}, \dots, X_{D,t})_{t \geq 1}$ where t is the discrete time index. In this study, we consider only a special class of DBNs, called 2-slice Temporal Bayesian Networks (2-TBN). A 2-TBN is a DBN which satisfies the

Markov property of order 1, that is, the future of the studied stochastic process is independent from its past given its present. This property is formally denoted by $\mathbf{X}_{t-1} \perp\!\!\!\perp \mathbf{X}_{t+1} | \mathbf{X}_t$ for all $t \geq 2$. A pair of BNs $(\mathcal{M}_1, \mathcal{M}_\rightarrow)$ allows to define a 2-TBNs. BN \mathcal{M}_1 represents the joint distribution of the initial process $\mathbf{X}_1 = (X_{1,1}, \dots, X_{D,1})$. According to (1), this distribution admits the following factorisation :

$$P(\mathbf{X}_1) = P(X_{1,1}, \dots, X_{D,1}) = \prod_{d=1}^D P(X_{d,1} | \mathbf{X}_{\text{pa}_{d,1}}),$$

where $\text{pa}_{d,1}$ is the indices sequence of the parents of the d -th variable in slice 1.

BN \mathcal{M}_\rightarrow defines the transition model, i.e. the distribution of \mathbf{X}_t given \mathbf{X}_{t-1} . In this case the BN factorisation leads to :

$$P(\mathbf{X}_t | \mathbf{X}_{t-1}) = P(X_{1,t}, \dots, X_{D,t} | X_{1,t-1}, \dots, X_{D,t-1}) = \prod_{d=1}^D P(X_{d,t} | \mathbf{X}_{\text{pa}_{d,t}}),$$

where $\text{pa}_{d,t}$ denotes the indices sequence of the parents of $X_{d,t}$ which can only stand in slices t and $t-1$ according to the Markov property of order 1.

Thereby, the joint distribution of the random variables over a temporal horizon T , i.e. $(\mathbf{X}_t)_{1 \leq t \leq T} = (X_{1,t}, \dots, X_{D,t})_{1 \leq t \leq T}$, can be found out by simply "unrolling" the 2-TBN until a sequence of length T is obtained :

$$P\left((\mathbf{X})_{1 \leq t \leq T}\right) = \prod_{d=1}^D P\left(X_{d,1} | X_{\text{pa}_{d,1}}\right) \prod_{t=2}^T \prod_{d=1}^D P\left(X_{d,t} | X_{\text{pa}_{d,t}}\right). \quad (2)$$

Note that as in the static BN formalism if all variables take their values in a finite and discrete domain, the underlying DBNs and 2-TBNs are also said to be discrete and finite (DF-DBNs and 2-DF-TBNs).

Besides, considering 2-TBNs as a pair of two static BNs allows to easily derive the previously mentioned learning procedures to both initial and transition models. On the other hand, the inference problem in 2-TBNs suffers from an important increase of time and space complexity. Specific algorithms have been proposed to partially resolve this issue [3, 15].

Remind that a 2-TBN verifies the Markov property of order 1. Therefore, inference in any temporal slice t is possible as soon as the distribution of the process in slice $t-1$ is known. Basically, only a subset of the process is sufficient to verify the Markov property. This particular subset is called the interface and corresponds to the variables that have children in the next slice. The interface is formally denoted by \mathbf{X}_{I_t} where I_t represents the indices sequence of the variables in slice t that have children in slice $t+1$. Thereby, the 2-TBN interface property can be written in the following manner :

$$\mathbf{X}_{t-1} \perp\!\!\!\perp \mathbf{X}_{t+1} | \mathbf{X}_{I_t}. \quad (3)$$

An inference method exploiting the interface property coupled with junction tree is proposed in [12]. Basically, any static inference algorithms can be used to update the interface distribution at each time step. This method is applied in section 4 with the elimination algorithm instead of the junction tree method.

3. Graphical Duration Models

2-TBNs are general tools allowing to model dynamic complex systems. Besides, it is important to notice that using a 2-TBNs to represent a variable depending on its own past is equivalent to use a Markov chain to describe its local transition model. Consequently, the sojourn-times associated to each variable state are geometrically distributed. Unfortunately this assumption turns out to be insufficient in most of practical systems.

To overcome this restriction, this article presents a specific 2-TBN structure, denoted Graphical Duration Model (GDM), able to represent dynamic multi-states systems having arbitrary complex sojourn-time distributions. The first contribution of this paper is to extend the variable duration models introduced in [12]. These models were originally devoted to resolve HMM-like learning problems by taking into account the information of the time spent in each model state. In this section both the graphical and the probabilistic description of GDMs are addressed.

3.1. Graphical description

The 2-TBN graphical structure of a GDM is depicted in Figure 1. The model consists of the following three variables : X_t denotes the system state at time t ; S_t denotes the remaining sojourn-time at time t in the current system state; Z_t denotes the context of the system. The GDM graphical structure shows that the current system state X_t depends on the previous system state X_{t-1} , the previous remaining duration S_{t-1} and the current context Z_t . On the other hand, the current remaining sojourn-time S_t is dependant on the sojourn-time S_{t-1} , the context Z_t , the current state X_t and the previous state X_{t-1} . Effectively, when using a GDM, the system is characterised by a state transition model, a sojourn-time model and a context model. Details about the probabilistic properties in GDMs are given in section 3.2.

According to the 2-TBN factorisation property (2), the joint probability distribution of $(Z_t, X_t, S_t)_{1 \leq t \leq T}$ over a sequence of length T is given by :

$$\begin{aligned}
 P((Z_t, X_t, S_t)_{1 \leq t \leq T}) &= P(Z_1)P(X_1|Z_1)P(S_1|X_1, Z_1) \cdot & (4) \\
 &\prod_{t=2}^T P(Z_t)P(X_t|X_{t-1}, S_{t-1}, Z_t) \times \\
 &P(S_t|X_{t-1}, S_{t-1}, Z_t, X_t).
 \end{aligned}$$

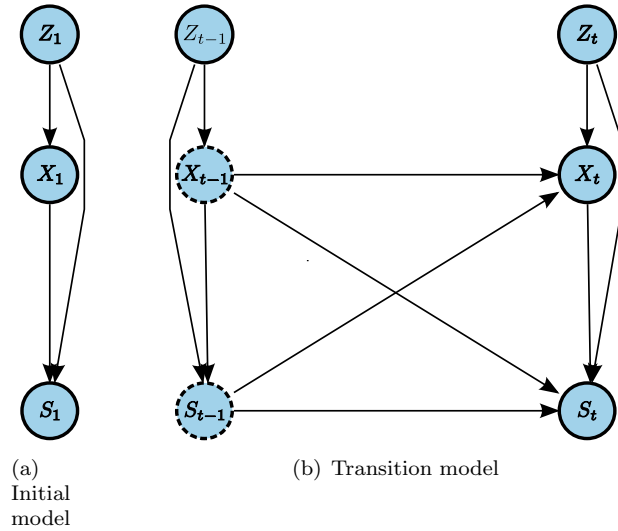


Figure 1: 2-TBN representation of a GDM. Z_t , X_t and S_t represent respectively the context, the system state and the remaining sojourn-time at time t . The figure 1(a) gives the GDM initial model \mathcal{M}_1 while the figure 1(b) depicts its transition model \mathcal{M}_\rightarrow . Dashed variables in figure 1(b) indicate the interface variables, i.e. $\mathbf{X}_{I_t} = (X_t, S_t)$.

By applying the interface definition to a GDM, it appears that (X_t, S_t) characterises the interface. Consequently, the interface property (3) leads to :

$$(Z_{t-1}, X_{t-1}, S_{t-1}, Z_t) \perp\!\!\!\perp (Z_{t+1}, X_{t+1}, S_{t+1}) | (X_t, S_t). \quad (5)$$

These two previous results are used in section 4 where a specific inference algorithm is described aiming to evaluate some useful probabilistic indicators concerning the studied system.

3.2. Probabilistic description

This section addresses the specifications of CPDs under the discrete and finite assumption. In other words, the question is to explain the role of each CPD involved in (4). The purpose of the CPDs description given in the sequel is to provide sufficient indications to allow fast implementations.

3.2.1. Context CPDs

In many applications the studied system can differently behave according to various contexts. Let \mathcal{Z} be the discrete and finite set containing the possible contexts. The following context distribution is defined for each $z \in \mathcal{Z}$ by :

$$P(Z_t = z) = \omega_{t,z}, \quad (6)$$

where $\omega_{t,z}$ is the probability that the system stands in the context z at time t . The distribution ω_t is then a sum-to-one vector of N_Z elements assuming N_Z denotes the number of considered contexts.

It is of interests to note that in practical applications the number of context variables can be greater than one. Basically, in a multivariate context case Z_t denotes a vector of random variables $(Z_{t,1}, \dots, Z_{t,M})$ taking their values in $\mathcal{Z} = \mathcal{Z}_1 \times \dots \times \mathcal{Z}_M$. The characterisation of ω_t given in (6) does not change. Nevertheless the larger is the number of context variables, the more space consuming becomes the distribution ω_t . A way to reduce the space complexity of the context distribution could be to use a Bayesian network to take advantage of the independence relationships among context variables.

3.2.2. Initial system state CPD

Let \mathcal{X} be the set of the N_X system states. The initial system state CPD allows to describe the initial system characteristics according to the context. In other words, for each context $z \in \mathcal{Z}$ and each state $x \in \mathcal{X}$:

$$P(X_1 = x | Z_1 = z) = V_{1,z,x},$$

where $V_{1,z,x}$ gives the probability for the system to start in state x given the context z . CPD V_1 is then a right stochastic matrix¹ consisting of N_Z rows and N_X columns.

3.2.3. Initial sojourn-time CPD

The initial sojourn-time CPD gives the distributions of the time spent in each possible (context,state) configuration. This CPD is defined over the discrete and finite domain $\mathcal{S} = \{1, 2, \dots, s, \dots, \geq N_S\}$, where N_S is the maximum number of steps the process can spend in any (context,state) configurations. The domain \mathcal{S} holds N_S elements of which the last one is not a single value, but a right-opened interval. This is a convention aiming to preserve both the infinite nature of time and the finite assumption of the CPD. This is formally written for each $z \in \mathcal{Z}$ and each $x \in \mathcal{X}$ as follows :

$$\begin{aligned} P(S_1 = s | Z_1 = z, X_1 = x) &= F_{1,z,x,s} \quad 1 \leq s < N_S \\ \text{and} & \\ P(S_1 \geq N_S | Z_1 = z, X_1 = x) &= F_{1,z,x,N_S} \end{aligned}, \quad (7)$$

where $F_{1,z,x,s}$ is the probability to spend s time units in the state x and in the context z . Thus, F_1 can be represented as a right stochastic matrix consisting of $N_Z N_X$ rows and N_S columns. N_S can be interpreted as a discrete-time truncation parameter. The larger N_S , the more accurate the representation of the sojourn-time distribution. On the other hand, choosing a too large N_S value will have immediate consequences on both the space needed to store the CPD, i.e. $N_Z N_X N_S$ values, and the time complexity of inference.

Note finally that the main advantage of this approach stems from the flexibility concerning the choice of the sojourn-time distributions. Indeed, any kind of shapes can be handled provided the discretisation parameter N_S has been

¹i.e. a matrix consisting of real numbers in $[0, 1]$ with each row summing to one.

appropriately chosen. Thereby a wide range of dynamic phenomena can be captured. A survey on discrete sojourn-time distributions is proposed in [4] giving interesting ideas to fill CPD F_1 .

3.2.4. System state transition CPD

The system state transition CPD aims to describe how the system goes from one state to another. Two cases have to be dealt with :

1. The sojourn-time in the previous state x' is up, i.e. $S_{t-1} = 1$. Therefore, a transition occurs at time t and a new current state x is drawn according to the following CPD :

$$P(X_t = x | X_{t-1} = x', S_{t-1} = 1, Z_t = z) = Q_{x',z,x}^{\text{sys}}$$

where $Q_{\cdot,z,\cdot}^{\text{sys}}$ is a right stochastic matrix consisting of N_X rows and N_X columns giving the probability that the system goes to the state x from the state x' given the context z . We refer to Q^{sys} as the static system transition matrix since it is time-independent.

2. It remains some time to be spent in the state x' , i.e. $S_{t-1} \geq 2$. In this case the system deterministically stays in the current state which is formally expressed by :

$$P(X_t = x | X_{t-1} = x', S_{t-1} \geq 2, Z_t = z) = I_{x',x} = \begin{cases} 1 & \text{if } x = x' \\ 0 & \text{otherwise} \end{cases} ,$$

where I is the identity matrix consisting of N_X rows and N_X columns, ensuring (i.e. with probability one) that the next state x will be equal to the previous state x' .

Finally, the global system state transition CPD is defined as :

$$P(X_t = x | X_{t-1} = x', S_{t-1} = s', Z_t = z) = Q_{x',s',z,x}^{\rightarrow} = \begin{cases} Q_{x',z,x}^{\text{sys}} & \text{if } s' = 1 \\ I_{x',x} & \text{if } s' \geq 2 \end{cases} . \quad (8)$$

3.2.5. Sojourn-time transition CPD

The sojourn-time transition CPD aims to update the remaining sojourn-time in the current state at each time step. This CPD is also used to provide a new sojourn-time when a transition occurs. As in the case of the system state transition CPD, the same two cases have to be considered :

1. If a transition occurs, i.e. $S_{t-1} = 1$, a new sojourn-time is selected according to the following CPD :

$$P(S_t = s | X_{t-1} = x', S_{t-1} = 1, Z_t = z, X_t = x) = F_{x',z,x,s}^{\text{sys}}$$

where F^{sys} is a right stochastic matrix consisting of $N_X^2 N_Z$ rows and N_S columns specifying the sojourn-time distributions given the current context, the previous system state and the new system state. The definition of

F^{sys} follows the same principle as the definition of F_1 . Moreover in some practical situations, F^{sys} can be considered independent on the previous state such that for each $x' \in \mathcal{X}$, $F_{x',:,:,}^{\text{sys}} = F_1$, where the colon symbol means the corresponding dimension is entirely considered.

2. The sojourn in the current state is not over, i.e. $S_{t-1} \geq 2$. The remaining sojourn-time $s' \geq 2$ has to be deterministically counted down as follows :

$$\begin{aligned} P(S_t = s | X_{t-1} = x', S_{t-1} = s', Z_t = z, X_t = x) &= C_{s',s} \\ &= \begin{cases} 1 & \text{if } s = s' - 1, \\ 0 & \text{otherwise} \end{cases}, \end{aligned}$$

where C is a right stochastic matrix consisting of $N_S - 1$ rows and N_S columns defined as the identity matrix with $N_S - 1$ rows concatenated with a zero-valued column on the right.

Finally, the global sojourn-time transition CPD is defined as

$$\begin{aligned} P(S_t = s | X_{t-1} = x', S_{t-1} = s', Z_t = z, X_t = x) &= F_{x',s',z,x,s}^{\rightarrow} \\ &= \begin{cases} F_{x',z,x,s}^{\text{sys}} & \text{if } s' = 1, \\ C_{s',s} & \text{if } s' \geq 2 \end{cases}. \end{aligned} \quad (9)$$

3.3. Space complexity of a GDM

Assuming that a GDM only consists of discrete and finite CPDs, the calculation of the corresponding space complexity is straightforward. Indeed, it suffices to count the number of elements defining each CPD characterising the GDM namely ω_t , V_1 , F_1 , Q^{\rightarrow} and F^{\rightarrow} . According to the descriptions and the notations introduced in section 3.2, the space complexity order for each CPD, denoted by $SC(\cdot)$, is given for the initial CPDs and the context CPD by :

$$SC(\omega_t) = \mathcal{O}(N_Z), \quad SC(V_1) = \mathcal{O}(N_Z N_X) \quad \text{and} \quad SC(F_1) = \mathcal{O}(N_Z N_X N_S).$$

Counting naively the number of elements needed to defined transition CPDs Q^{\rightarrow} and F^{\rightarrow} leads to the following space complexities :

$$SC^{\text{naive}}(Q^{\rightarrow}) = \mathcal{O}(N_Z N_X^2 N_S), \quad \text{and} \quad SC^{\text{naive}}(F^{\rightarrow}) = \mathcal{O}(N_Z N_X^2 N_S^2).$$

Using an *ad hoc* storage for CPDs Q^{\rightarrow} and F^{\rightarrow} allows to significantly reduce their space complexity. Indeed, exploiting the piecewise definitions (8) and (9) and applying a sparse storage technique to matrices I and C lead to the following *ad hoc* space complexities :

$$SC^{\text{ad hoc}}(Q^{\rightarrow}) = SC(Q^{\text{sys}}) + SC(I) = \mathcal{O}(N_Z N_X^2) + \mathcal{O}(N_X) = \mathcal{O}(N_Z N_X^2),$$

and,

$$SC^{\text{ad hoc}}(F^{\rightarrow}) = SC(F^{\text{sys}}) + SC(C) = \mathcal{O}(N_Z N_X^2 N_S) + \mathcal{O}(N_S) = \mathcal{O}(N_Z N_X^2 N_S).$$

The space complexity of a GDM is the sum of the previous complexities. As a result, the space needed to store a GDM is then on the order of $\mathcal{O}(N_Z N_X^2 N_S)$ elements.

4. Inference

This section addresses the problem of the probabilistic inference in a dynamic system represented by means of a GDM. Remind that Z_t , X_t and S_t denote respectively the random variables associated to the context, the state and the remaining sojourn-time at time t .

The purpose of this section is to evaluate the probability of any system state trajectories of length T . In other words, the underlying inference problem boils down to the calculation of the following probability :

$$\tau(\mathcal{E}) = \tau(\mathcal{E}_1, \dots, \mathcal{E}_T) = P(X_1 \in \mathcal{E}_1, \dots, X_T \in \mathcal{E}_T), \quad (10)$$

where each $\mathcal{E}_t \subseteq \mathcal{X}$, $1 \leq t \leq T$ is a subset of system states. According to the choice of $\mathcal{E} = (\mathcal{E}_t)_{1 \leq t \leq T}$ the probability $\tau(\mathcal{E})$ allows to extract various information related to the system behaviour over time. Let now define the function $\Psi_t^{\mathcal{E}}$ for each $x \in \mathcal{X}$ and each $s \in \mathcal{S}$ given a fixed trajectory \mathcal{E} as :

$$\Psi_{t,x,s}^{\mathcal{E}} = P(X_1 \in \mathcal{E}_1, \dots, X_{t-1} \in \mathcal{E}_{t-1}, X_t = x, S_t = s). \quad (11)$$

Under the discrete and finite assumption, $\Psi_t^{\mathcal{E}}$ is a matrix consisting of N_X rows and N_S columns. Note besides that summing (11) over \mathcal{E}_t leads to the value of interests $\tau(\mathcal{E})$. The following proposition gives a recursive characterisation of $\Psi_t^{\mathcal{E}}$ based on the derivation of the elimination method presented in [6].

Proposition 1 (Recursive characterisation of $\Psi_t^{\mathcal{E}}$). *Let $(Z_t, X_t, S_t)_{t \geq 1}$ be the sequence of random variables associated to a GDM and taking their values in domains \mathcal{Z} , \mathcal{X} and \mathcal{S} respectively (see section 3). Consider also a subset collection of \mathcal{X} denoted by $(\mathcal{E}_1, \dots, \mathcal{E}_t)$. Then for any $t \geq 1$, $\Psi_t^{\mathcal{E}}$ can be recursively expressed for each $x \in \mathcal{X}$ and each $s \in \mathcal{S}$,*

$$\Psi_{t,x,s}^{\mathcal{E}} = \begin{cases} \Psi_{1,x,s} & \text{if } t = 1 \\ \sum_{x' \in \mathcal{E}_{t-1}} \sum_{s'=1}^{N_S} \Psi_{t-1,x',s'}^{\mathcal{E}} \sum_{z \in \mathcal{Z}} \omega_{t,z} Q_{x',s',z,x}^{\rightarrow} F_{x',s',z,x,s}^{\rightarrow} & \text{if } t \geq 2 \end{cases} \cdot \quad (12)$$

The proof of this proposition is given in appendix A. The previous result represents a generic exact inference method to compute $\Psi_t^{\mathcal{E}}$ based on the naive storage of CPDs Q^{\rightarrow} and F^{\rightarrow} . As far as we know, most of Bayesian network implementations use similar approaches to perform this kind of calculations. As it is described in [1], the time complexity, denoted by $TC(\cdot)$, of sum-product calculation involving discrete and finite CPDs is equivalent to the space complexity of the largest intermediate term processed during the summations. In (12) this largest term is $\omega_t \cdot Q^{\rightarrow} \cdot F^{\rightarrow}$. The order of the time complexity associated to this naive computation of $\Psi_t^{\mathcal{E}}$ is then :

$$TC^{\text{naive}}(\Psi_t^{\mathcal{E}}) = SC(\omega_t \cdot Q^{\rightarrow} \cdot F^{\rightarrow}) = SC^{\text{naive}}(F^{\rightarrow}) = \mathcal{O}(N_Z N_X^2 N_S^2).$$

The following proposition represents the second contribution of this paper. It provides a fast recursive scheme aiming to compute $\Psi_t^{\mathcal{E}}$.

Proposition 2 (Fast recursive characterisation of $\Psi_t^\mathcal{E}$). Let $(Z_t, X_t, S_t)_{t \geq 1}$ be the sequence of random variables associate to a GDM and taking their values in domains \mathcal{Z}, \mathcal{X} and \mathcal{S} respectively (see section 3). Consider also a collection of subsets of \mathcal{X} denoted by $\mathcal{E} = (\mathcal{E}_1, \dots, \mathcal{E}_t)$. Then for any $t \geq 1$, $\Psi_t^\mathcal{E}$ is recursively defined for each $x \in \mathcal{X}$ and each $s \in \mathcal{S}$ by :

$$\Psi_{t,x,s}^\mathcal{E} = \begin{cases} \sum_{z \in \mathcal{Z}} \omega_{1,z} V_{1,z,x} F_{1,z,x,s} & \text{if } t = 1 \\ \sum_{x' \in \mathcal{E}_{t-1}} \Psi_{t-1,x',1}^\mathcal{E} \sum_{z \in \mathcal{Z}} \omega_{t,z} Q_{x',z,x}^{\text{sys}} F_{x',z,x,s}^{\text{sys}} + \tilde{\Psi}_{t-1,x,s}^\mathcal{E} & \text{if } t \geq 2 \end{cases}, \quad (13)$$

$$\text{where } \tilde{\Psi}_{t,x,s}^\mathcal{E} = \begin{cases} \Psi_{t,x,s+1}^\mathcal{E} & \text{if } x \in \mathcal{E}_t \text{ and } 1 \leq s < N_S \\ 0 & \text{otherwise} \end{cases}.$$

The proof of the proposition is given in appendix A. This proposition refines the result obtained in Proposition 1. It relies on the exploitation of the piecewise definition of Q^\rightarrow and F^\rightarrow . In addition, the deterministic behaviour of Q^\rightarrow and F^\rightarrow when $S_t \geq 2$ (no transition) is also used to simplify the calculations. The largest term involves in the summations appearing in (13) is $\omega \cdot Q^{\text{sys}} \cdot F^{\text{sys}}$. Consequently, the order of the time complexity associated to the *ad hoc* computation of $\Psi_t^\mathcal{E}$ described in Proposition 2 is then

$$TC^{\text{ad hoc}}(\Psi_t^\mathcal{E}) = SC(\omega_t \cdot Q^{\text{sys}} \cdot F^{\text{sys}}) = SC(F^{\text{sys}}) = \mathcal{O}(N_Z N_X^2 N_S).$$

This result leads to Algorithm 1 allowing to compute $\tau(\mathcal{E})$ for any state trajectory \mathcal{E} . The time complexity of this algorithm arises essentially from the computation of $\Psi_t^\mathcal{E}$. Consequently, the time complexity associated to Algorithm 1 is on the order of $\mathcal{O}(TN_Z N_X^2 N_S^2)$ using Proposition 1 as against $\mathcal{O}(TN_Z N_X^2 N_S)$ using Proposition 2. This shows the relevancy of the proposed inference method since it leads to the reduction of the time complexity of an order of magnitude in N_S . This is particularly interesting since N_S is often large for the reasons mentioned in section 3.2.3.

Algorithm 1 Iterative algorithm to compute any state trajectories of a system represented with GDM.

Require: A GDM verifying the probabilistic description presented in section 3.2; a state trajectory $\mathcal{E} = (\mathcal{E}_1, \dots, \mathcal{E}_T)$.

Ensure: Compute $\tau(\mathcal{E}) = P(X_1 \in \mathcal{E}_1, \dots, X_T \in \mathcal{E}_T)$.

- 1: Compute $\Psi_1^\mathcal{E} = \sum_{z \in \mathcal{Z}} \omega_{1,z} V_{1,z,x} F_{1,z,x,s}$
 - 2: Compute $\tau(\mathcal{E}_1) = \sum_{x \in \mathcal{E}_1} \sum_{s=1}^{N_S} \Psi_{1,x,s}^\mathcal{E}$
 - 3: **for** $t = 2$ to T **do**
 - 4: Compute $\Psi_{t,x,s}^\mathcal{E}$ according to Proposition 2
 - 5: Compute $\tau(\mathcal{E}_1, \dots, \mathcal{E}_t) = \sum_{x \in \mathcal{E}_t} \sum_{s=1}^{N_S} \Psi_{t,x,s}^\mathcal{E}$
 - 6: **end for**
-

5. Lifetime analysis using GDMs

5.1. Preliminaries

Assume the set of system states \mathcal{X} is partitioned into two sets \mathcal{X}_A and \mathcal{X}_D (i.e. $\mathcal{X} = \mathcal{X}_A \cup \mathcal{X}_D$ with $\mathcal{X}_A \cap \mathcal{X}_D = \emptyset$), respectively for alive states and for dead states.

Let $S : \mathbb{N}^* \mapsto [0, 1]$ denotes the discrete-time survival function defined as :

$$S(t) = P(X_1 \in \mathcal{X}_A, \dots, X_t \in \mathcal{X}_A), \quad t \in \mathbb{N}^*. \quad (14)$$

$S(t)$ represents the probability that the system is always stayed in an alive state until moment t . It is of interests to notice the link between the evaluation of the survival function and Algorithm 1. Indeed, $S(t)$ can be found out by setting $\mathcal{E}_1 = \mathcal{X}_A, \dots, \mathcal{E}_t = \mathcal{X}_A$ as inputs of the algorithm.

5.2. Case study description

Consider the case of a clinical trial aimed to evaluate the effect of two treatments on a given incurable disease. The disease is supposed to have three phases : regular, fast and death. The objective is then to exploit clinical trial results in order to quantify the lifetime gain for each treatment. In this application the studied system is symbolized by an affected patients. A patient can be either in the regular phase (R) or in the fast phase (F), or dead (D) with $\mathcal{X}_D = \{D\}$. The medical treatment represents the system context which can be either the first treatment (T1) or the second treatment (T2), or no treatment (T0). Assume now that the theoretical disease progression is governed for each treatment by the transition rates presented in Table 1 and the sojourn-time distributions given in Table 2. Table 1 confirms that the considered pathology is incurable.

	no treatment	treatment 1	treatment 2
regular \rightarrow fast	20%	50%	90%
regular \rightarrow dead	80%	50%	10%
fast \rightarrow dead	100%	100%	100%

Table 1: Transition rates characterising the disease state changes for each treatment.

An infected patient always ends up hitting the fast phase of the disease and then the dead state. However, treatment 1 makes infected patients go through the intermediate fast phase in 50% of the cases. Treatment 2 is even better since infected patients go directly to the dead state only in 10% of the cases.

Table 2 shows that treatment 1 and 2 allows to decrease γ , and thus the progression speed of the infection.

Note that in practice the previous information are unknown. However, a clinical trial consists in following a population of independent infected patients from the disease discovery to the death. This population is divided in three

	no treatment	treatment 1	treatment 2
regular	$\mathcal{W}(20, 3)$	$\mathcal{W}(20, 1)$	$\mathcal{W}(20, 0.85)$
fast	$\mathcal{W}(10, 3)$	$\mathcal{W}(10, 1)$	$\mathcal{W}(10, 0.85)$
dead	∞	∞	∞

Table 2: Sojourn-time distributions (in weeks) associated to each disease phases and each treatment. $\mathcal{W}(\mu, \gamma)$ denotes the Weibull distribution with scale parameter μ and shape parameter γ . It is worth reminding that a Weibull distribution with scale parameter $\gamma = 1$ is equivalent to the exponential distribution of parameter μ .

groups containing the same number of subjects. In order to observe the effect of each treatment, group T0 is not treated, group T1 is treated with treatment 1 and group T2 is treated with treatment 2. The clinical trial results consist for each group of the number of occurrences of each transition and the number of occurrences of each sojourn-time. The objective is to evaluate the survival function according to each group from these data.

In other to show the relevancy of our approach, two disease progression models are built. The first one relies on GDMs and the second one relies on Markov chains (MCs). Both models are supposed to be learnt from the clinical trial data using the classic maximum likelihood estimation method [9].

5.2.1. Results

Algorithm 1 is used to compute the survival function corresponding to both the GDM and MC representations. This is made possible since MCs can be considered as a special case of the GDM framework by removing the sojourn-time variable. The theoretical estimation is obtained using Monte-Carlo simulations. Figure 2(a) shows the survival functions associated to patients without treatment computed from the theoretical model, the GDM model and the MC model. In this case, both the GDM and MC models have been learnt from 100 patients. Figures 2(b) and 2(c) provide similar results for treatment T1 and treatment T2 respectively. Table 3 gives a summary of the Mean Relative Absolute Error (MRAE) resulting of the computation of the survival function from both the GDM and MC models. The comparisons are carried out with the theoretical evaluation of the survival function. The average error percentage at time t is given for each amount of available data and each treatment.

These results show that the GDM approach is particularly interesting when the studied system is governed by non-exponentially distributed sojourn-times. This is the case of the disease progression under treatments T0 and T2. Indeed, the treatment T0 leads to a theoretical progression verifying Weibull-distributed sojourn-times with scale parameter $\gamma = 3$. This situation is far from the exponentially distributed sojourn-time hypothesis implicitly assumed by the MC approach. This explains the large MRAE made by the MC modelling in this context. Besides, as the MC model is clearly false for treatment T0, increas-

# of patients	Treatment	GDM MRAE (%)	MC MRAE (%)	Ratio
10	T0	28.33	234.93	8.29
100	T0	11.61	191.73	16.51
1000	T0	2.40	192.40	80.29
10	T1	21.6	7.30	0.34
100	T1	10.7	5.44	0.51
1000	T1	2.79	2.39	0.85
10	T2	7.97	6.66	0.83
100	T2	4.73	4.85	1.02
1000	T2	1.01	4.71	4.65

Table 3: Mean Relative Absolute Error (MRAE) between theoretical and estimated survival functions. The last column denotes the ratio between the MC MRAE and the GDM MRAE.

ing the number of data does not necessarily improve the quality of the survival function. For treatment T2, the scale parameter is closer from the exponential situation. Consequently, the MRAE associated to the MC assumption is lower. When few data are available, the MC approach appears to be more robust than the GDM approach. However, the GDM modelling quickly provides better results as soon as the number of data increases. Finally, the treatment T1 denotes the only context in which the MC approach is more relevant. In this case, the MC approach fits to the theoretical model. This explains the better survival function estimation. These comparisons show that the GDM approach is more appropriate as soon as the exponentially distributed sojourn-time assumption is not realistic. Remark that it is often the case when the question concerns the representation of complex systems.

Besides, it is worth noting that various other interesting indicators can be deduced from these results (e.g. the mean time to death, the hazard rate, the mean residual lifetime. . .). Further analyses can then be performed to decide if one of the treatments is significantly better than another.

6. Conclusions

An original approach is described in this article aiming to represent a wide range of discrete duration models based on the DBN framework. Indeed, a specific DBN structure is proposed referred to as Graphical Duration Models (GDMs). Both its graphical structure and its probabilistic properties are studied. Then the GDM features such as the multi-state modelling, the complex handling of sojourn-time distributions and the context consideration are detailed.

Besides, a fast inference algorithm devoted to efficiently compute the probability of any system state trajectories is given along with its proof of correctness. In addition, a survival analysis problem is presented to illustrate the suitability

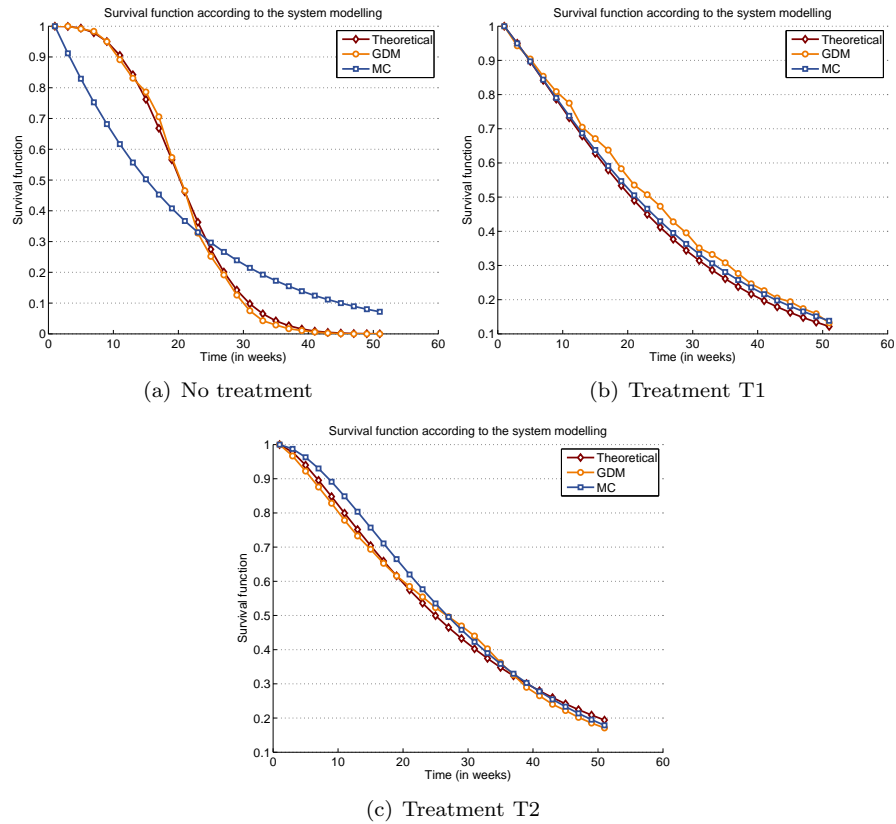


Figure 2: Comparison between theoretical and estimated survival functions.

of GDMs compared to commonly used approaches such as Markov chain modelling. The method is successfully applied to evaluate the lifetime function of patients under different treatments, legitimating the use of the GDMs to deal with practical problems. In addition, it is worth emphasizing on the fact that the proposed work is generic and could be applied to other domains in which transitions and durations modelling are the matter of concerns (e.g. economy, meteorology, ...).

Works are in progress to extend the GDM structure and the associate inference algorithm by introducing action variables. The objective is to model systems in which an action might be selected modifying the future behaviour of the system at each time step. The second perspective of interests is to consider the long-run behavior of a system represented with a GDM, and hence to establish under what conditions a GDM possesses a steady distribution.

References

- [1] M. Arias and F. J. Díez. Operating with potentials of discrete variables. *International Journal of Approximate Reasoning*, 46(1):166–187, 2007.
- [2] H. Boudali and J. B. Dugan. A discrete-time Bayesian network reliability modeling and analysis framework. *Reliability Engineering & System Safety*, 87(3):337–349, March 2005.
- [3] X. Boyen and D. Koller. Tractable Inference for Complex Stochastic Processes. In *Uncertainty in artificial Intelligence*. pages 32–42. Madison, Wisconsin, 1998.
- [4] C. Bracquemond and O. Gaudoin. A survey on discrete lifetime distributions. *International Journal on Reliability, Quality, and Safety Engineering*, 10(1):69–98, 2003.
- [5] D. R. Cox. Regression models and life-tables. *Journal of the Royal Statistical Society.*, 34(2):187–220, 1972.
- [6] R. Dechter. Bucket elimination: A unifying framework for reasoning. *Artificial Intelligence*, 113(1-2):41–85, 1999.
- [7] Fabio Gagliardi Cozman. Generalizing variable elimination in Bayesian networks. In *the IBERAMIA/SBIA Workshops on Probabilistic Reasoning in Artificial Intelligence*, pages 27–32. Editora Tec Art, 2000.
- [8] Finn V. Jensen. *An introduction to Bayesian networks*. UCL Press, 1996.
- [9] M. Jordan. *Learning in graphical models*. MIT Press, Cambridge, MA, USA, 1999.
- [10] J. D. Kalbfleisch and R. L. Prentice. *The Statistical Analysis of Failure Time Data. Second Edition*. Wiley Series in Probability and Statistics. Wiley, 2002.
- [11] S. L. Lauritzen and D. J. Spiegelhalter. Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society*, 50(2):157–224, 1988.
- [12] Kevin P. Murphy. *Dynamic Bayesian Networks : Representation, Inference and Learning*. PhD thesis, University of California, Berkeley, 2002.
- [13] J. Pearl. *Probabilistic Reasoning in Intelligent Systems : Networks of Plausible Inference*. Morgan Kaufmann, 1988.
- [14] Philippe Weber and Lionel Jouffe. Reliability modelling with dynamic Bayesian networks. In *5th IFAC Symposium on fault Detection, Supervision and Safety of Technical Processes*, Washington D.C., USA, June 2003.

- [15] G. Zweig. *A forward-backward algorithm for inference in Bayesian networks and an empirical comparison with HMMs*. Master thesis, Department of Computer Science, U.C. Berkeley, 1996.

A. Proof of Propositions 1 and 2

PROOF (OF PROPOSITION 1). Let proceed by induction. For $t = 1$, the result is straightforward since :

$$\Psi_1^{\mathcal{E}} = \sum_{Z_1} P(Z_1, X_1, S_1) = P(X_1, S_1) = \Psi_1.$$

Assume $\Psi_{t-1}^{\mathcal{E}} = P(X_1 \in \mathcal{E}_1, \dots, X_{t-2} \in \mathcal{E}_{t-2}, X_{t-1}, S_{t-1})$ for $t \geq 2$. Using simple probability manipulation rules, we have

$$\begin{aligned} \Psi_t^{\mathcal{E}} &= \sum_{(Z_\tau, S_\tau)_{1 \leq \tau \leq t-1}, Z_t} P\left((Z_\tau, X_\tau \in \mathcal{E}_\tau, S_\tau)_{1 \leq \tau \leq t-1}, Z_t, X_t, S_t\right) \\ &= \sum_{(Z_\tau, S_\tau)_{1 \leq \tau \leq t-1}, Z_t} P\left(Z_t, X_t, S_t \mid (Z_\tau, X_\tau \in \mathcal{E}_\tau, S_\tau)_{1 \leq \tau \leq t-1}\right) \times \\ &\quad P\left((Z_\tau, X_\tau \in \mathcal{E}_\tau, S_\tau)_{1 \leq \tau \leq t-1}\right) \\ \Psi_t^{\mathcal{E}} &= \sum_{S_{t-1}, Z_t} P(Z_t, X_t, S_t \mid X_{t-1} \in \mathcal{E}_{t-1}, S_{t-1}) \times \\ &\quad \sum_{(Z_\tau, S_\tau)_{1 \leq \tau \leq t-2}, Z_{t-1}} P\left((Z_\tau, X_\tau \in \mathcal{E}_\tau, S_\tau)_{1 \leq \tau \leq t-1}\right). \end{aligned}$$

The last result comes from the GDM interface property depicted in equation (5). Besides, let now remark that :

$$\sum_{(Z_\tau, S_\tau)_{1 \leq \tau \leq t-2}, Z_{t-1}} P\left((Z_\tau, X_\tau \in \mathcal{E}_\tau, S_\tau)_{1 \leq \tau \leq t-1}\right) = P\left((X_\tau \in \mathcal{E}_\tau)_{1 \leq \tau \leq t-1}, S_{t-1}\right).$$

Then, the computation of $\Psi_t^{\mathcal{E}}$ can be pursued by using the GDM factorisation property (4) as follows :

$$\begin{aligned} \Psi_t^{\mathcal{E}} &= \sum_{S_{t-1}, Z_t} P(Z_t, X_t, S_t \mid X_{t-1} \in \mathcal{E}_{t-1}, S_{t-1}) P\left((X_\tau \in \mathcal{E}_\tau)_{1 \leq \tau \leq t-1}, S_{t-1}\right) \\ &= \sum_{S_{t-1}, Z_t} P(Z_t) P(X_t \mid X_{t-1} \in \mathcal{E}_{t-1}, S_{t-1}, Z_t) \times \\ &\quad P(S_t \mid X_{t-1} \in \mathcal{E}_{t-1}, S_{t-1}, Z_t, X_t) P\left((X_\tau \in \mathcal{E}_\tau)_{1 \leq \tau \leq t-1}, S_{t-1}\right). \end{aligned}$$

This can be equivalently rewritten for each $x \in \mathcal{X}$ and each $s \in \mathcal{S} = \{1, \dots, N_S\}$ by :

$$\begin{aligned}\Psi_{t,x,s}^{\mathcal{E}} &= \sum_{x' \in \mathcal{E}_{t-1}} \sum_{s'=1}^{N_S} \sum_{z \in \mathcal{Z}} P(Z_t = z) P(X_t = x | X_{t-1} = x', S_{t-1} = s', Z_t = z) \times \\ &\quad \underbrace{P(S_t = s | X_{t-1} = x', S_{t-1} = s', Z_t = z, X_t = x) \times P(X_1 \in \mathcal{E}_1, \dots, X_{t-2} \in \mathcal{E}_{t-2}, X_{t-1} = x', S_{t-1} = s')}_{=\Psi_{t-1,x',s'} \text{ by assumption}} \\ \Psi_{t,x,s}^{\mathcal{E}} &= \sum_{x' \in \mathcal{E}_{t-1}} \sum_{s'=1}^{N_S} \sum_{z \in \mathcal{Z}} \omega_{t,z} Q_{x',s',z,x}^{\rightarrow} F_{x',s',z,x,s}^{\rightarrow} \Psi_{t-1,x',s'}.\end{aligned}$$

PROOF (OF PROPOSITION 2). Begin with the result of Proposition 1 :

$$\Psi_{t,x,s}^{\mathcal{E}} = \sum_{x' \in \mathcal{E}_{t-1}} \sum_{s'=1}^{N_S} \Psi_{t-1,x',s'} \sum_{z \in \mathcal{Z}} \omega_{t,z} Q_{x',s',z,x}^{\rightarrow} F_{x',s',z,x,s}^{\rightarrow}, \quad t \geq 2.$$

Then split the sum over the previous sojourn-time domain between $s' = 1$ and $s' \geq 2$, that is for each $x \in \mathcal{X}$ and each $s \in \mathcal{S}$,

$$\begin{aligned}\Psi_{t,x,s}^{\mathcal{E}} &= \sum_{x' \in \mathcal{E}_{t-1}} \Psi_{t-1,x',1}^{\mathcal{E}} \sum_{z \in \mathcal{Z}} \omega_{t,z} \underbrace{Q_{x',1,z,x}^{\rightarrow}}_{=Q_{x',z,x}^{\text{sys}}} \underbrace{F_{x',1,z,x,s}^{\rightarrow}}_{=F_{x',z,x,s}^{\text{sys}}} + \\ &\quad \sum_{x' \in \mathcal{E}_{t-1}} \sum_{s' \geq 2} \Psi_{t-1,x',s'}^{\mathcal{E}} \sum_{z \in \mathcal{Z}} \omega_{t,z} \underbrace{Q_{x',s',z,x}^{\rightarrow}}_{=\mathbb{I}(x'=x)} \underbrace{F_{x',s',z,x,s}^{\rightarrow}}_{=\mathbb{I}(s'=s+1)}.\end{aligned}$$

The indicator functions introduced when $s' \geq 2$ allow to make disappear both the sums over x' and s' . Besides, the sum over z makes ω_t sum to one since ω_t does not depend on x' neither on s' .

$$\begin{aligned}\Psi_{t,x,s}^{\mathcal{E}} &= \sum_{x' \in \mathcal{E}_{t-1}} \Psi_{t-1,x',1}^{\mathcal{E}} \sum_{z \in \mathcal{Z}} \omega_{t,z} Q_{x',z,x}^{\text{sys}} F_{x',z,x,s}^{\text{sys}} + \\ &\quad \sum_{s' \geq 2} \mathbb{I}(s' = s+1) \underbrace{\sum_{x' \in \mathcal{E}_{t-1}} \Psi_{t-1,x',s'}^{\mathcal{E}} \mathbb{I}(x' = x)}_{=\mathbb{I}(x \in \mathcal{E}_{t-1}) \Psi_{t-1,x,s'}^{\mathcal{E}}} \underbrace{\sum_{z \in \mathcal{Z}} \omega_{t,z}}_{=1} \\ &= \mathbb{I}(x \in \mathcal{E}_{t-1}) \mathbb{I}(1 \leq s \leq N_S - 1) \mathcal{E}_{t-1} \Psi_{t-1,x,s+1}^{\mathcal{E}}\end{aligned}$$

Remark that $\mathbb{I}(x \in \mathcal{E}_{t-1}) \mathbb{I}(1 \leq s \leq N_S - 1) \mathcal{E}_{t-1} \Psi_{t-1,x,s+1}^{\mathcal{E}}$ is exactly $\tilde{\Psi}_{t-1,x,s}^{\mathcal{E}}$ as defined in Proposition 2. Hence, the result follows.