

## **USING PROCESS MINING TO MODEL INTERARRIVAL TIMES: INVESTIGATING THE SENSITIVITY OF THE ARPRA FRAMEWORK**

Niels Martin  
Benoît Depaire

An Caris

Hasselt University  
Faculty of Business Economics  
Agoralaan Building D  
3590 Diepenbeek, BELGIUM

Hasselt University / Research Foundation Flanders  
(FWO)  
Faculty of Business Economics  
Agoralaan Building D / Egmontstraat 5  
3590 Diepenbeek / 1000 Brussels, BELGIUM

### **ABSTRACT**

Accurately modeling the interarrival times (IAT) is important when constructing a business process simulation model given its influence on process performance metrics such as the average flow time. To this end, the use of real data from information systems is highly relevant as it becomes more readily available. This paper considers event logs, a particular type of file containing process execution information, as a data source. To retrieve an IAT input model from event logs, the recently developed ARPRA framework is used, which is the first algorithm that explicitly integrates the notion of queues. This paper investigates ARPRA's sensitivity to the initial parameter set estimate and the size of the original event log. Experimental results show that (i) ARPRA is fairly robust for the specification of the initial parameter estimate and (ii) ARPRA's output represents reality more closely for larger event logs than for smaller logs.

### **1 INTRODUCTION**

Business process simulation (BPS) refers to the imitation of business process behavior through the use of a simulation model (Melão and Pidd 2003). By mimicking the real system, simulation can identify the effects of operational changes prior to implementation and contribute to the analysis and improvement of business processes (Melao and Pidd 2003).

A BPS model consists of several building blocks such as activities and entities, where the latter refers to dynamic objects that flow through the system and on which activities are executed (Kelton Sadowski and Zupick 2015), e.g. customers of a car rental company. As with all BPS model components, modelling tasks are associated to entities (Martin Depaire and Caris 2014). This paper is related to entity arrival rate modeling, i.e. defining the pattern according to which entities arrive at the process.

Given its influence on process performance metrics such as the average flow time, i.e. the total time in the system, accurately modeling the entity arrival process is crucial. To identify an interarrival time (IAT) input model, i.e. a parameterized probability distribution (Henderson 2003) for the time between the arrivals of consecutive entities, inputs can be gathered by e.g. observing the process. However, as process observations are time-consuming, using more readily available information sources is useful. In this respect, process execution information which is automatically recorded by process-aware information systems (PAIS), such as ERP systems, can be useful. This information is recorded as events, e.g. the registration of a customer, in files called event logs. These minimally contain an ordered set of events for each case, where a case is the event log equivalent for an entity. However, additional information is

typically recorded for each event such as a timestamp expressing its time of occurrence and the associated resource. Information extraction from event logs belongs to the process mining field (van der Aalst 2011).

The limited work on the use of process mining to retrieve an IAT input model from an event log assumes that entities arrive at their first recorded timestamp. Hence, IAT observations can be directly calculated from the log and a probability distribution can be fitted. However, as event logs typically only record activity execution information, the aforementioned assumption is not appropriate when queues can be formed for the first activity. The latter is the case when the first activity of the process has a non-zero duration and limited resources. Queue formation will cause entities to arrive sooner than the time related to the first recorded event, e.g. the start of the first activity’s execution. When correspondence between entity arrival and the first recorded timestamp is wrongfully assumed, an inaccurate IAT input model is obtained (Martin Depaire and Caris 2015).

To retrieve an IAT input model from an event log while taking into account queue formation, a novel algorithm called ARPRA is developed. The algorithm has been shown to provide a significantly more accurate IAT input model than a benchmark approach ignoring queue formation (Martin Depaire and Caris 2015). To render an executable algorithm, several parameters need to be specified. To evaluate the influence of those settings on ARPRA’s performance, this paper presents a sensitivity analysis. More specifically, ARPRA’s sensitivity is investigated for (i) the initial estimate for the IAT input model parameters and (ii) the event log size.

The remainder of this paper is structured as follows. The following section introduces a running example and presents the scarce related work. The third section outlines ARPRA and its current operationalization. In the fourth section, ARPRA’s sensitivity for the initial parameter estimate and log size is investigated. The paper ends with a conclusion.

## 2 BACKGROUND

### 2.1 Running example

Throughout this paper, the process of a fictitious car rental company will serve as a running example. The simplified process model is visualized in Figure 1.

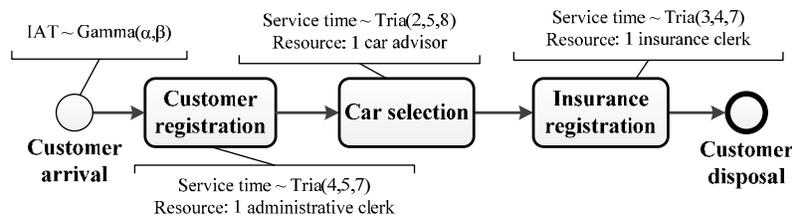


Figure 1: Running example.

The car rental company’s process can be described as follows. Upon arrival, a customer is registered by inserting all relevant information in the company’s PAIS. When all required information is recorded, the customer will select an appropriate car together with the car advisor. The final process step is the creation of insurance documents. Based on limited process observations, the company assumes that service times follow a triangular distribution. All assumed parameters are annotated in Figure 1, with minutes as the time unit and resource capacities constant throughout the day.

The running example can be used to illustrate the consequences of inaccurate IAT modeling. Suppose real customer IATs are gamma distributed, which is a more generic distribution than the popular exponential distribution (Law 2007), with 1.50 and 5.00 as its shape and scale parameter. Table 1 presents some process performance metrics for deviating parameter sets. E.g. a 10% overestimation of the distribution parameters leads to an underestimation of the average waiting time for ‘Customer registration’ by 54.15%, showing the need to carefully model IATs when building a BPS model.

Table 1: Effect of inaccurate IAT modeling.

Shape / scale parameter	Average flow time	Average waiting time for 'Customer registration'	Utilization administrative clerk
1.50 / 5.00 (real)	21.27	4.58	0.731
1.65 / 5.50 (+10%)	18.27 (-14.10%)	2.10 (-54.15%)	0.615 (-15.87%)
1.35 / 4.50 (-10%)	26.27 (+23.51%)	9.21 (+101.10%)	0.842 (+15.18%)

To estimate an IAT input model, this paper uses an event log as it typically contains a large volume of process execution information. An extract from an example event log is given in Table 2. Each line in the log corresponds to a single event, e.g. the first line refers to the start of activity 'Customer registration' for customer with identification code 83 by administrative clerk Sue. Note that identification codes are assumed to be unique for each visit. The activity's corresponding end event is recorded as the second event in Table 2. Given the focus on IAT modeling, the events associated to the first activity of the process, i.e. 'Customer registration', will be of key interest in the remainder of this paper.

Table 2: Extract from example event log.

Unique customer ID	Timestamp	Activity	Event type	Resource
...	...	...	...	...
83	11/06/2015 09:17:38	Customer registration	start	Administrative clerk: Sue
83	11/06/2015 09:21:57	Customer registration	end	Administrative clerk: Sue
83	11/06/2015 09:26:04	Car selection	start	Car advisor: Mike
84	11/06/2015 09:31:21	Customer registration	start	Administrative clerk: Sue
83	11/06/2015 09:35:42	Car selection	end	Car advisor: Mike
83	11/06/2015 09:37:09	Insurance registration	start	Insurance clerk: Dave
...	...	...	...	...

## 2.2 Related work

Despite the importance of having an accurate IAT input model and the increasing presence of information systems generating event logs, thorough research on how event logs can support IAT modeling is lacking.

A preliminary insight in the arrival rate can be gathered using a dotted chart, a graphical representation marking the events for each case over time using dots (Song and van der Aalst 2007). By analyzing the position of the first dot for consecutive entities, the arrival rate can be explored. However, a visual inspection is insufficient to obtain an IAT input model. The closest to our work is Rozinat et al. (2009), which is the only reference on process mining in a BPS context that briefly mentions arrival rate modeling. They define the IAT as the difference between the first recorded timestamp of two consecutive cases. After IAT calculation, an a priori assumed exponential distribution is fitted to these observations.

Both dotted charts and the approach of Rozinat et al. (2009) implicitly assume that a case arrives at its first recorded timestamp, which is not necessarily true. As highlighted in the introduction, queue formation for the first activity causes a discrepancy between entity arrival and the first activity start event, where the latter is the best proxy for entity arrival when only activity execution information is logged. Hence, the implicit assumption made in literature is only appropriate when entities do not have to queue upon arrival, i.e. (i) when the first activity has a service time of zero or (ii) when the associated resources are idle or have an unlimited capacity. Besides under these particular circumstances, queuing behavior is relevant.

As ignoring queue formation can bias the obtained IAT input model, an algorithm is lacking which considers first activity queueing during the model estimation process. To this end, ARPRA is introduced in Martin, Depaire and Caris (2015), which is the first algorithm that explicitly takes into account queue

formation. It has been shown to outperform an IAT modeling method that ignores queue formation. Further details on the algorithm are provided in Section 3.

### 3 OVERVIEW OF ARPRA

This section gives an overview of ARPRA, the Arrival Rate Parameter Retrieval Algorithm. The functioning of the ARPRA framework, visualized in Figure 2, can be summarized as follows. All ARPRA’s inputs originate from an event log, with the percentage of entities that queued upon arrival ( $q$ ) as its main input. Given this percentage, the algorithm iteratively adjusts the parameter set ( $\Psi$ ) of a particular IAT probability distribution ( $\tilde{f}$ ) until the queue percentage in a simulated log ( $\tilde{q}$ ) matches the queue proportion from the original event log ( $q$ ). When this is the case, the parameter set ( $\Psi$ ) is recorded and the algorithm proceeds to its next iteration until a pre-specified number of such matches ( $r$ ) are obtained. Afterwards, an aggregated parameter estimate is returned ( $\Psi_{selected}$ ).

Even though ARPRA is a generic framework, the remainder of this section details the operationalization suggested in Martin, Depaire and Caris (2015) as an implementation is required to investigate ARPRA’s sensitivity. For the sake of clarity and brevity, only key concepts are included.

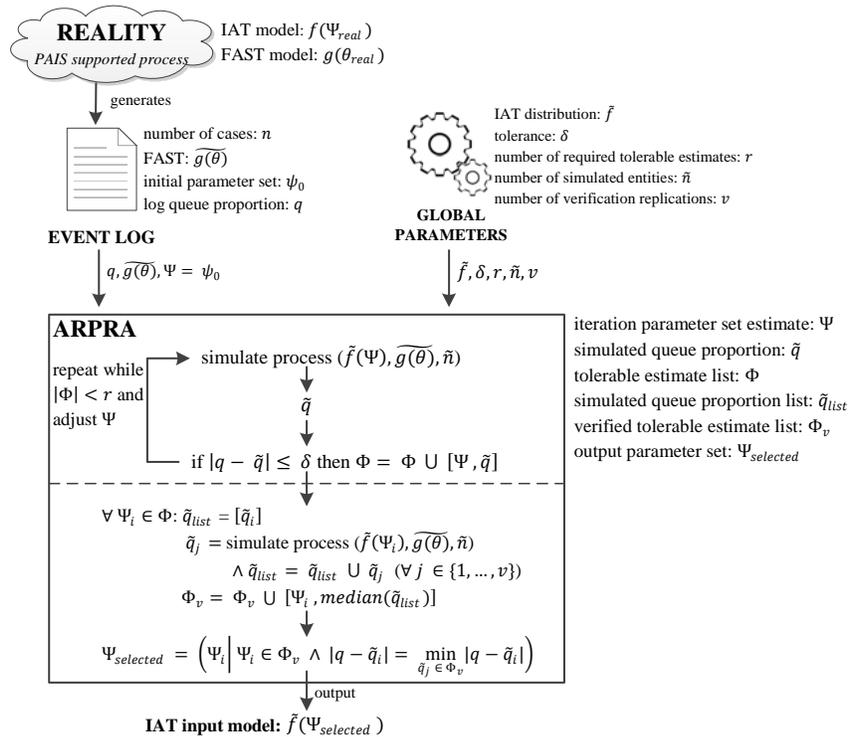


Figure 2: Overview of the generic ARPRA framework (Martin Depaire and Caris 2015).

#### 3.1 Event log inputs

At the top of the schematic overview in Figure 2, a PAIS-supported process is shown with a particular, unknown, IAT input model  $f(\Psi_{real})$  and first activity service time (FAST) probability distribution  $g(\theta_{real})$ . This process generates an event log from which three inputs for ARPRA are retrieved: the percentage of entities that queued upon arrival ( $q$ ), an initial parameter set estimate for the IAT input model ( $\psi_0$ ) and knowledge on the service time for the first activity ( $\widehat{g}(\theta)$ ).

Firstly, to mine  $q$  from the event log, the first activity timestamps are investigated for consecutive entities. Consider Table 3, where the first activity start and end timestamps are recorded for the first four

customers in the car rental company’s event log. A customer had to wait for activity ‘Customer registration’ when the execution starts immediately after ‘Customer registration’ ended for the previous customer. When this is the case, e.g. for customer 3 in Table 3, the value True is assigned to a boolean *Queue*. Otherwise, this value is set to False. After assigning *Queue* values to all customers,  $q$  corresponds to the percentage of customers for which *Queue* = True.

Table 3: Illustration of *Queue*-value assignment.

Unique customer ID	‘Customer registration’ start timestamp	‘Customer registration’ end timestamp	Queue
1	07/04/2015 09:17:38	07/04/2015 09:22:57	False
2	07/04/2015 09:33:04	07/04/2015 09:37:19	False
3	07/04/2015 09:37:19	07/04/2015 09:42:01	True
4	07/04/2015 09:42:01	07/04/2015 09:47:50	True
...	...	...	...

Secondly, an initial parameter set ( $\psi_0$ ), which is used in ARPRA’s first iteration, is retrieved from the log. This involves fitting probability distribution  $\tilde{f}$ , specified in Section 3.2, on known IAT values in the original event log. IATs are known when two consecutive entities did not queue upon arrival, i.e. have *Queue* = False. When this is the case, the start timestamp for this activity corresponds to entity arrival. Hence, exact IATs can be calculated, e.g. for customer 2 in Table 3.

Finally, knowledge on the first activity service time ( $\widehat{g(\theta)}$ ) from the event log is required as it will be used during the construction of a simulated log. As both the start and end timestamp are assumed to be recorded, service times can directly be calculated. Service time observations from the original log are directly used when ARPRA creates a simulated log in its iterations. This approach is selected as queues are formed influenced by the interaction between arrival and service times. Consequently, ARPRA will maintain the order of service time observations from the original event log. When the size of ARPRA’s simulated log ( $\tilde{n}$ ) exceeds the size of the original event log ( $n$ ), the observed FAST sequence is repeated.

### 3.2 Global parameters

Besides the event log inputs, global parameters need to be defined to obtain an executable algorithm. An IAT probability distribution ( $\tilde{f}$ ) needs to be put forward for which a parameter set ( $\Psi$ ) will be retrieved by ARPRA. In the current operationalization,  $\tilde{f}$  is equated to a gamma distribution: a probability distribution with shape parameter  $\alpha$  and scale parameter  $\beta$ , i.e.  $\Psi = \{\alpha, \beta\}$ . It is purposefully selected because, when  $\alpha = 1$ , a gamma distribution corresponds to an exponential distribution (Law 2007). Consequently, the gamma distribution is more generic, but still allows for the often cited exponential IAT probability distribution (Law 2007, Kelton, Sadowski, and Zupick 2015).

Besides  $\tilde{f}$ , four other global parameters should be specified: (i) the tolerated deviation from  $q$  within ARPRA’s iterations  $\delta$  is set to 0.01, (ii) the size of the simulated log  $\tilde{n}$  should equal 400, (iii) the number of tolerable estimates required to end the algorithm  $r$  is 10 and (iv)  $v$ , the number of additional iterations to verify the stability of the queue proportion for a given tolerable parameter set, also equals 10. The meaning of these settings is detailed in Section 3.3.

### 3.3 Parameter retrieval using ARPRA

Using the above event log inputs and global parameters, ARPRA retrieves an IAT input model. The rectangle marked with ARPRA in Figure 2 is subdivided in two parts by a dashed line. The upper part reflects the identification of a series of tolerable parameter sets, while the lower part outlines final output selection.

An IAT input model, obtained using  $\tilde{f}$  and  $\psi_0$ , initiates the phase in which tolerable parameter sets are identified. Given  $\tilde{f}(\psi_0)$ , the process is simulated to create a simulated event log. From this log, the queue proportion ( $\tilde{q}$ ) is calculated as outlined for the original log in Section 3.1. When  $\tilde{q}$  is within a tolerance margin  $\delta$  from the queue proportion in the original log ( $q$ ), the parameter set  $\Psi$  is added to  $\Phi$  and iteration proceeds. Conversely, when  $\tilde{q}$  does not have a tolerable value, the algorithm continues to the next iteration without recording  $\Psi$ . To adjust parameter set  $\Psi = \{\alpha, \beta\}$  across iterations, the observation that the gamma distribution mean  $\mu$  equals  $\alpha\beta$  is used (Law 2007). The mean IAT can be estimated using the original event log by considering the time between the first activity start timestamps for the first and last entity in the log. Dividing this timespan by the number of arrivals in this period provides an approximation for  $\mu$ . As  $\mu$  fixes the relationship between both parameters, adjusting  $\Psi$  is brought down to varying  $\tilde{\alpha}$  and changing  $\tilde{\beta}$  using  $\tilde{\beta} = \mu/\tilde{\alpha}$ . The adjustment of  $\tilde{\alpha}$  across iterations occurs as follows:

- When  $\tilde{q} > q + \delta$  for the current iteration, too many entities have been queueing. Consequently,  $\tilde{\alpha}$  is increased for the following iteration as this increases the mean IAT for a given shape parameter. The adjustment size is determined by applying a percentage increase to  $\tilde{\alpha}$  corresponding to the percentage point deviation between  $\tilde{q}$  and  $q$ . However, given the non-linear relationship between  $\tilde{\alpha}$  and  $\tilde{q}$ , the calculated adjustment is smoothed downward to avoid too large adjustments. E.g. a result of 0.008 leads to an actual increase in  $\tilde{\alpha}$  of only 0.001.
- When  $\tilde{q} < q - \delta$  for the current iteration, too many entities have been queueing, requiring a decrease in  $\tilde{\alpha}$  as this reduces the mean IAT for a given shape parameter. The adjustment size is defined analogously to the previous situation.
- When  $|q - \tilde{q}| \leq \delta$ ,  $\Psi$  is added to  $\Phi$ . In order to explore the entire range of parameter sets that leads to tolerable queue proportions, a large adjustment in  $\tilde{\alpha}$  is required to push  $\tilde{q}$  outside the range  $[q - \delta ; q + \delta]$  for the following iteration. The direction of this adjustment is determined by the value of  $\tilde{q}$  for the current and two prior iterations. If  $\tilde{q} > q$  in the current iteration,  $\tilde{\alpha}$  is doubled for the next iteration to reduce  $\tilde{q}$ , except when  $\tilde{q} < q$  for the two prior iterations. In the latter case,  $\tilde{\alpha}$  is halved to explore another parameter region. The inverse holds if  $\tilde{q} < q$  in the current iteration. When  $\tilde{q} = q$  for the current iteration,  $\tilde{q}$  values for the three prior iterations are taken into account, where the third lag is a tie-breaker.

ARPRA's iteration process ends when a pre-specified number of parameter sets are recorded, i.e. when  $|\Phi| = r$ . When this is the case, ARPRA proceeds to the lower part of the rectangle in Figure 2 to select its final output. For each of the  $r$  parameter sets in  $\Phi$ ,  $v$  additional  $\tilde{q}$  values are determined. This is done because each simulated log is based on random IAT draws from  $\tilde{f}(\Psi)$ . As a consequence, different  $\tilde{q}$  values can be obtained for the same parameter set  $\Psi$ . Hence, for each  $\Psi \in \Phi$ ,  $v$  additional queue proportions are determined and the median  $\tilde{q}$  value will be used in the remainder of the algorithm. Note that, in the current implementation,  $r = v = 10$ .

From the previous, a list  $\Phi_v$  containing parameter estimates with verified  $\tilde{q}$  values is obtained. From this list, the final output of ARPRA is selected. Parameter set  $\Psi_{selected}$  is returned, which is the recorded parameter set that leads to the closest approximation of  $q$ . In case of ties, an aggregated parameter set is returned by e.g. calculating the mean for  $\tilde{\alpha}$  and its associated  $\tilde{\beta}$  using  $\tilde{\beta} = \mu/\tilde{\alpha}$ .

#### 4 SENSITIVITY OF ARPRA

The outline of ARPRA in Section 3 only considers a single specification of the event log inputs and global parameters. To assess the influence of alternative definitions on ARPRA's performance, a sensitivity analysis is conducted. This paper focuses on (i) the initial parameter set  $\psi_0$  and (ii) the size of the original event log  $n$ . The first subsection outlines the experimental design. Afterwards, the results of the sensitivity analysis for  $\psi_0$  and  $n$  are reported in the second and third subsection, respectively.

## 4.1 Experimental design

The sensitivity analysis evaluates the influence of alternative definitions of  $\psi_0$  and  $n$  on ARPRA's ability to rediscover a known IAT input model solely using an event log. To this end, the car rental company's process, introduced in Section 2.1, is the starting point. Given the focus on IAT modeling, especially the first activity 'Customer registration' is of interest. Within this context, a real IAT input model  $f(\Psi_{real}) = \text{Gamma}(\alpha, \beta)$  is put forward, which is of course unknown in reality. Using this IAT probability distribution, an event log is created. Only using this log, ARPRA is applied and a particular output  $\tilde{f}(\Psi_{selected}) = \text{Gamma}(\tilde{\alpha}, \tilde{\beta})$  is obtained. The deviation between  $\Psi_{real}$  and  $\Psi_{selected}$  reflects the ability of the algorithm to rediscover the real IAT input model based on an event log. To put the observed deviations into perspective, the percentage deviation from the real value is considered. For a particular real IAT input model, several definitions for  $\psi_0$  or  $n$  are considered. By comparing the percentage deviations for these alternative definitions, their influence on ARPRA's performance can be evaluated.

The experiment described above is repeated for several IAT gamma distributions representing reality. These distributions are randomly defined by drawing values for  $\alpha$  and  $\beta$  from a uniform distribution over a particular range. To specify these ranges, it should be taken into account that the service times for the activity 'Customer registration' follow a triangular distribution with parameters (4, 5, 7) minutes. Hence, the mean service time equals 5.33 minutes. To allow 'Customer registration' to reach a steady state, its utilization factor should be smaller than one (Hillier and Lieberman 2010). This implies that the mean IAT,  $\alpha\beta$  for the gamma distribution, should be at least 5.33 minutes as the activity only has a single server. Consequently, when real parameter values need to be drawn,  $\alpha$  is drawn from a uniform distribution between 1 and 2 and  $\beta$  from a uniform distribution between 5.33 and 7.

## 4.2 Sensitivity for initial parameter estimate ( $\psi_0$ )

### 4.2.1 Considered specifications

The initial parameter estimate defines the IAT input model that will be used in the first iteration of ARPRA. To verify the sensitivity for  $\psi_0$ , three distinct definitions are considered:

- Definition 1: the initial value of  $\alpha$  is set to 1 and the initial  $\beta$  equals the mean IAT, where the mean IAT is derived from the event log as outlined in Section 3;
- Definition 2: the initial values for  $\alpha$  and  $\beta$  are determined by fitting a gamma distribution on entities for which IATs can be calculated in the original event log, which is the  $\psi_0$  definition used in Section 3. When less than five exact IATs are retrievable from the log,  $\alpha$  is set to 1 and the initial  $\beta$  equals the mean IAT.
- Definition 3: the initial  $\alpha$  is set to the mean IAT and the initial  $\beta$  is set to 1.

Using the experimental design in Section 4.1, a real IAT input model is specified and an event log is created. Solely using this log, ARPRA is applied three times with alternative  $\psi_0$  definitions. All other specifications are retained from the operationalization outlined in Section 3. The experiment is repeated for 500 distinct real IAT input models in experiment 1A. In experiment 1B, an additional varying factor is introduced as 150 real values of  $\alpha$  are drawn from each of three distinct ranges for  $\alpha$  and  $\beta$ .

### 4.2.2 Results

For experiment 1A, 500 experiments are executed following the above specifications. For each of the real IAT input models, an event log is created. The queue proportion  $q$ , a key input for ARPRA, in the 500 event logs has a mean value of 50.93%. Minimum and maximum  $q$  values equal 18.75% and 95.50%.

To gain insight in the accuracy differences of ARPRA across  $\psi_0$  definitions, pairwise comparisons are required. For each pair of  $\psi_0$  definitions, a paired t-test for both IAT gamma distribution parameters is conducted on the absolute values of the percentage deviations from the real value, abbreviated by

$|\tilde{\kappa}_{ARPR} - \kappa_{real}|%$  with kappa representing  $\alpha$  or  $\beta$ . Absolute values are used as mainly the size of the deviation is deemed relevant, independent of whether it is an over- or underestimation. A Bonferroni correction is applied to correct for multiple comparisons, where a family-wise error rate of 5% is used. As a consequence, the three individual paired t-tests are evaluated at a 1.66% significance level (Blann 2015). The results of the pairwise comparisons are summarized in Figure 3 using a visualization method originating from Demšar (2006), where the  $m$  and  $s$  digits under each definition represent the mean  $|\kappa_{ARPR} - \tilde{\kappa}_{real}|%$  and its standard deviation, respectively. In Figure 3,  $\psi_0$  definitions are ranked by increasing  $|\tilde{\kappa}_{ARPR} - \kappa_{real}|%$ . The paired t-test tests the null hypothesis that the mean  $|\tilde{\kappa}_{ARPR} - \kappa_{real}|%$  using both  $\psi_0$  specifications is the same. Pairwise comparisons for which insufficient evidence is available to reject this null hypothesis are marked using bold line segments. The results show that the second definition for  $\psi_0$  leads to the lowest  $|\tilde{\kappa}_{ARPR} - \kappa_{real}|%$  for both  $\alpha$  and  $\beta$ . However, none of the performance differences are statistically significant. Hence, ARPRA is fairly robust for the initial parameter set  $\psi_0$ .

Besides the pairwise comparisons, the three  $\psi_0$  definitions can be evaluated independently by verifying if a bias is present in the obtained IAT gamma distribution parameters. To this end, the null hypothesis that the mean percentage deviation from the real parameter, denoted by  $(\tilde{\kappa}_{ARPR} - \kappa_{real})%$ , equals zero is evaluated at significance level 5%. Table 4, which expresses percentages in decimal form, shows that insufficient evidence is present to reject this null hypothesis for  $\alpha$ , regardless of the  $\psi_0$  definition used. For  $\beta$ , the null hypothesis can be rejected for all  $\psi_0$  specifications, with the second definition rendering the lowest mean  $(\tilde{\kappa}_{ARPR} - \kappa_{real})%$  for both  $\alpha$  and  $\beta$ .

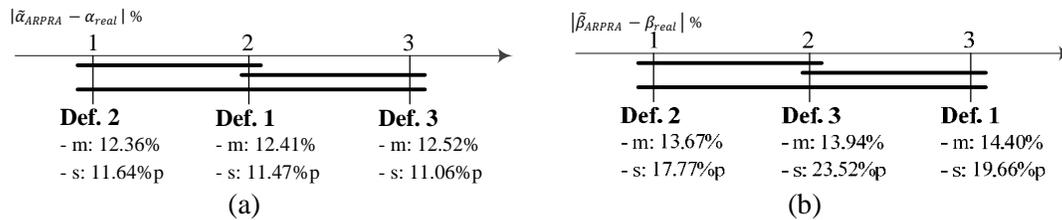


Figure 3: Pairwise comparison of  $|\tilde{\kappa}_{ARPR} - \kappa_{real}|%$  for (a)  $\alpha$  and (b)  $\beta$  (experiment 1A).

Table 4:  $(\tilde{\kappa}_{ARPR} - \kappa_{real})%$  results for alternative  $\psi_0$  definitions (experiment 1A).

$\psi_0$ def.	$(\tilde{\alpha}_{ARPR} - \alpha_{real})%$			$(\tilde{\beta}_{ARPR} - \beta_{real})%$		
	Mean value	St. dev. in sample	95% conf. int.	Mean value	St. dev. in sample	95% conf. int.
1	-0.0028	0.1690	[-0.0177,0.0120]	0.0404	0.2404	[0.0193,0.0615]
2	0.0127	0.1695	[-0.0022,0.0276]	0.0217	0.2232	[0.0021,0.0413]
3	0.0067	0.1671	[-0.0080,0.0214]	0.0302	0.2718	[0.0063,0.0541]

The results reported above suggest that ARPRA’s performance is relatively robust for the selected  $\psi_0$  definition. However, the effect of the  $\psi_0$  specification might be influenced by the real parameter set. To gain deeper understanding on this matter, experiment 1B is conducted, where three ranges for the real value of  $\alpha$  and  $\beta$  are defined: (i) Range 1:  $\alpha \in [1.0, 1.33] \wedge \beta \in [5.5, 6.0]$ , (ii) Range 2:  $\alpha \in [1.33, 1.66] \wedge \beta \in [6.0, 6.5]$  and (iii) Range 3:  $\alpha \in [1.66, 2.0] \wedge \beta \in [6.5, 7.0]$ . From each range, 150 values are drawn using a uniform distribution to represent the real IAT input model. The mean  $q$  values for the resulting event logs for range 1, 2 and 3 equal 78.56%, 46.61% and 27.95%, respectively.

Pairwise comparisons of  $|\tilde{\kappa}_{ARPR} - \kappa_{real}|%$  do not show statistically significant differences between the  $\psi_0$  definitions when real values are drawn from ranges 1 or 3, when a 5% family-wise error rate is used. These results are not visualized due to space limitations. Conversely, for real values from range 2, ARPRA is significantly more accurate with the second  $\psi_0$  definition than with the third specification for  $\alpha$  and  $\beta$ .

The results of the independent evaluation of the  $\psi_0$  definitions in each of the ranges are shown in Table 5. The null hypothesis states, once again, that the mean  $(\tilde{\kappa}_{ARPR} - \kappa_{real})\%$  equals zero. Independent from the  $\psi_0$  specification, insufficient evidence is present to reject this null hypothesis when the real values originate from ranges 1 and 3. Conversely, for real values from range 2, a small overestimation of  $\alpha$  is shown for all initial parameter definitions and a small underestimation of  $\beta$  for definitions 2 and 3.

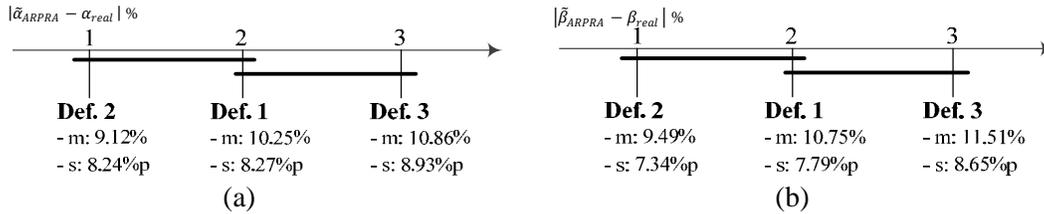


Figure 4: Pairwise comparison of  $|\tilde{\kappa}_{ARPR} - \kappa_{real}|\%$  for (a)  $\alpha$  and (b)  $\beta$  (experiment 1B, range 2)

Table 5:  $(\tilde{\kappa}_{ARPR} - \kappa_{real})\%$  results for alternative  $\psi_0$  definitions (experiment 1B).

$\psi_0$ def.	$(\tilde{\alpha}_{ARPR} - \alpha_{real})\%$			$(\tilde{\beta}_{ARPR} - \beta_{real})\%$		
	Mean value	St. dev. in sample	95% conf. int.	Mean value	St. dev. in sample	95% conf. int.
<b>Range 1: <math>\alpha \in [1.0, 1.33] \wedge \beta \in [5.5, 6.0]</math></b>						
1	0.0296	0.5575	[-0.0603,0.1195]	0.1018	0.4833	[0.0238,0.1798]
2	0.0446	0.3501	[-0.0119,0.1011]	0.0946	0.6494	[-0.0102,0.1994]
3	0.1512	1.2938	[-0.0576,0.3599]	0.1222	0.7407	[0.0027,0.2417]
<b>Range 2: <math>\alpha \in [1.33, 1.66] \wedge \beta \in [6.0, 6.5]</math></b>						
1	0.0276	0.1291	[0.0068,0.0484]	-0.0151	0.1322	[-0.0364,0.0063]
2	0.0311	0.1191	[0.0119,0.0503]	-0.0213	0.1183	[-0.0404,-0.0022]
3	0.0407	0.1349	[0.0189,0.0624]	-0.0255	0.1421	[-0.0484,-0.0026]
<b>Range 3: <math>\alpha \in [1.66, 2.0] \wedge \beta \in [6.5, 7.0]</math></b>						
1	0.0043	0.1071	[-0.0130,0.0216]	0.0068	0.1164	[-0.0120,0.0256]
2	0.0018	0.1034	[-0.0149,0.0184]	0.0083	0.1107	[-0.0095,0.0261]
3	0.0031	0.1102	[-0.0147,0.0208]	0.0082	0.1145	[-0.0102,0.0267]

In summary, experiments show that ARPR is fairly robust for the specification of the initial parameter set. Even though definition 2 tends to lead to the most accurate results, performance differences are not statistically significant. Only when the real parameters have medium values, i.e. in range 2, ARPR’s output is significantly more accurate when  $\psi_0$  is based on definition 2 than on definition 3.

### 4.3 Sensitivity for the size of the event log ( $n$ )

#### 4.3.1 Considered specifications

ARPR is mainly guided by  $q$ , which is retrieved from the original event log. The event log is generated by a process with a particular IAT and FAST input model. As the log only contains events for a limited period of time, its size influences the likelihood that the arrival and service time patterns in the event log are representative for their equivalents in reality. The larger the original event log, the more likely it becomes that this is the case. As a consequence, it is useful to investigate the sensitivity of ARPR to variations in the size of the original event log  $n$ .

Based on the experimental design outlined in Section 4.1, ARPR’s sensitivity to  $n$  is evaluated by considering five different log sizes, i.e.  $n \in \{100, 200, 400, 800, 1600\}$ . All other specifications, including the one for  $\psi_0$ , are retaken from the discussion in Section 3. This experiment is repeated 500

times. The entire procedure outlined to this point is executed twice: once with  $\tilde{n} = 400$  in experiment 2A, i.e. the specification mentioned in Section 3, and once with  $\tilde{n} = n$  in experiment 2B.

### 4.3.2 Results

Given the outlined experimental design, two series of 500 experiments are conducted. In each series, the 500 real parameter sets lead to the creation of 2500 event logs, for which  $q$  has a mean value of 48.31% for experiment 2A and 48.84% for experiment 2B. Minimum and maximum values equal 12% and 98% for experiment 2A and 11% and 96.75% for experiment 2B, respectively.

The results of the pairwise comparison of  $|\tilde{\kappa}_{ARPRA} - \kappa_{real}| \%$  across the different log sizes for experiment 2A are visualized in Figure 5. When a family-wise error rate of 5% is used and, i.e. a 0.5% significance level is applied to individual paired t-tests, results differ between  $\alpha$  and  $\beta$ . For  $\alpha$ , the accuracy of ARPRA significantly increases as the log size increases. Accuracy improvements can also be observed for  $\beta$  as  $n$  increases, but these are statistically insignificant for the comparison between (i)  $n = 400$  and  $n = 800$  and (ii)  $n = 800$  and  $n = 1600$ . A reason for the observed accuracy improvement is that larger the event log, the more representative  $q$  will be for the real arrival rate.

Besides the comparative assessment, ARPRA can also be evaluated by considering all log sizes independently. To this end, the null hypothesis that the mean  $(\tilde{\kappa}_{ARPRA} - \kappa_{real}) \%$  equals zero is evaluated for each value of  $n$  at the 5% significance level. The results are summarized in Table 6. Regarding  $\alpha$ , the mean and standard deviation of  $(\tilde{\kappa}_{ARPRA} - \kappa_{real}) \%$  tends to decrease as  $n$  increases. However, only for log sizes 800 and 1600 insufficient evidence is present to reject the aforementioned null hypothesis. For  $\beta$ , a similar null hypothesis cannot be rejected for all values of  $n$ , apart from 100. The standard deviation of  $(\tilde{\kappa}_{ARPRA} - \kappa_{real}) \%$  decreases as  $n$  increases for  $\beta$ . Regarding the mean value of  $(\tilde{\kappa}_{ARPRA} - \kappa_{real}) \%$ , a less consistent pattern is observable.

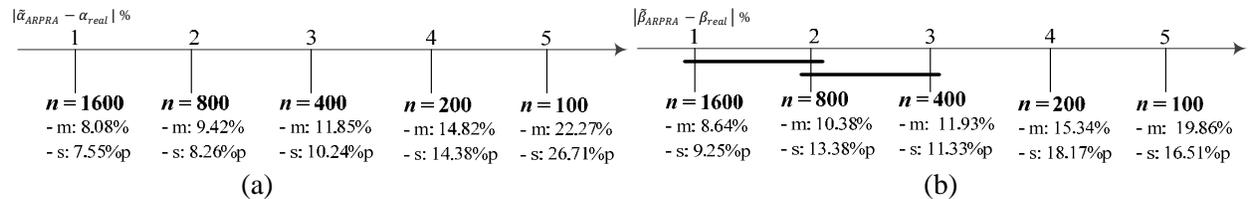


Figure 5: Pairwise comparison of  $|\tilde{\kappa}_{ARPRA} - \kappa_{real}| \%$  for (a)  $\alpha$  and (b)  $\beta$  (experiment 2A).

Table 6:  $(\tilde{\kappa}_{ARPRA} - \kappa_{real}) \%$  results for varying log sizes, with  $\tilde{n} = 400$  (experiment 2A).

Log size $n$	$(\tilde{\alpha}_{ARPRA} - \alpha_{real}) \%$			$(\tilde{\beta}_{ARPRA} - \beta_{real}) \%$		
	Mean value	St. dev. in sample	95% conf. int.	Mean value	St. dev. in sample	95% conf. int.
100	0.1073	0.3309	[0.0783, 0.1364]	-0.037	0.2558	[-0.0591, -0.0141]
200	0.0427	0.2021	[0.0249, 0.0604]	-0.004	0.2379	[-0.0253, 0.0165]
400	0.0267	0.1543	[0.0131, 0.0402]	-0.004	0.1644	[-0.0186, 0.0103]
800	0.0013	0.1254	[-0.0097, 0.0123]	0.0148	0.1687	[-0.0000, 0.0296]
1600	0.0026	0.1106	[-0.0071, 0.0123]	0.0096	0.1263	[-0.0015, 0.0207]

In experiment 2A,  $\tilde{n}$  equals 400 independent of the size of the original event log. Even though this limits the required computational efforts when the original log is large, it also implies that not all first activity service times from the original event log are used when  $n > \tilde{n}$ . Consequently, the number of simulated entities is equated to the size of the event log in experiment 2B. From the pairwise comparison of  $|\tilde{\kappa}_{ARPRA} - \kappa_{real}| \%$  across  $n$  values, visualized in Figure 6, it follows that the differences in accuracy between  $n = 1600$  and  $n = 800$  for  $\alpha$  and  $n = 400$  and  $n = 200$  for  $\beta$  become statistically insignificant.

When comparing the means and standard deviations in Figure 6 with the annotations of Figure 5, few noteworthy differences are observed. Hence, the effect of not using all first activity service times on ARPRA’s accuracy seems to be rather limited.

The results of the independent evaluation of log sizes are presented in Table 7. Insufficient evidence is available to reject the null hypothesis that the mean  $(\tilde{\kappa}_{ARPR\alpha} - \kappa_{real})\%$  equals zero at the 5% significance level for all log sizes for  $\beta$  and only for the largest event log for  $\alpha$ .

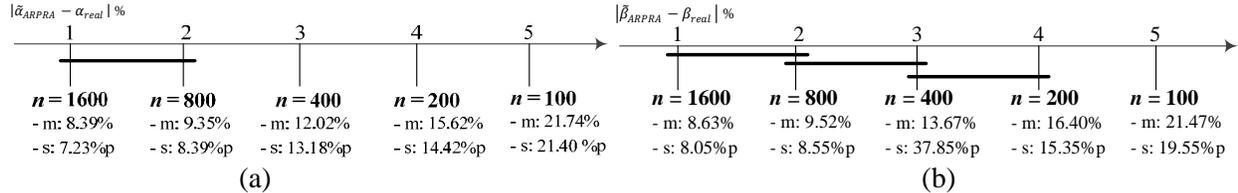


Figure 6: Pairwise comparison of  $|\tilde{\kappa}_{ARPR\alpha} - \kappa_{real}| \%$  for (a)  $\alpha$  and (b)  $\beta$  (experiment 2B).

Table 7:  $(\tilde{\kappa}_{ARPR\alpha} - \kappa_{real})\%$  results for varying log sizes, with  $\tilde{n} = n$  (experiment 2B).

Log size $n$	$(\tilde{\alpha}_{ARPR\alpha} - \alpha_{real})\%$			$(\tilde{\beta}_{ARPR\beta} - \beta_{real})\%$		
	Mean value	St. dev. in sample	95% conf. int.	Mean value	St. dev. in sample	95% conf. int.
100	0.0591	0.2994	[0.0328, 0.0854]	0.0078	0.2904	[-0.0177, 0.0333]
200	0.0261	0.2110	[0.0075, 0.0446]	0.0143	0.2243	[-0.0055, 0.0341]
400	0.0270	0.1765	[0.0115, 0.0425]	0.0135	0.4022	[-0.0218, 0.0489]
800	0.0231	0.1235	[0.0122, 0.0340]	-0.0079	0.1278	[-0.0191, 0.0034]
1600	0.0081	0.1106	[-0.0016, 0.0178]	0.0032	0.1181	[-0.0072, 0.0135]

To sum up, experimentation shows that the size of the original event log influences ARPRA’s performance. When increasing  $n$ , ARPRA’s output becomes significantly more accurate for  $\alpha$  when  $\tilde{n} = 400$ . For  $\beta$ , performance differences for successive increases in  $n$  become statistically insignificant from  $n = 400$  onwards. Moreover, the mean  $(\tilde{\kappa}_{ARPR\alpha} - \kappa_{real})\%$  tends to decrease as the size of the event log increases. Insufficient evidence was present to show a systematic bias for both parameters in large event logs. In larger event logs, the value of  $q$ , the main input of ARPRA, is more likely to be representative for the true underlying IAT input model. Consequently, ARPRA uses a more accurate guiding value for  $q$  and, hence, its output is more likely to approximate reality more closely. The influence of equating  $\tilde{n}$  to the size of the original event log is rather limited. When interpreting the results outlined above, it has to be taken into account that rather conservative values for  $n$  are used. Real-life event logs tend to contain events related to a large number of cases.

## 5 CONCLUSION

This paper presented a sensitivity analysis of ARPRA, an algorithm that derives an IAT input model from an event log. It is the first algorithm that explicitly takes the notion of queues into account, which is important as ignoring queuing behavior can bias the retrieved arrival patterns from the log. To execute ARPRA, several event log inputs and global parameters need to be defined. In order to assess the influence of those specifications on ARPRA’s performance, a sensitivity analysis is conducted. More specifically, the influence of (i) the initial estimate for the parameter set  $\psi_0$  and (ii) the size of the original event log  $n$  are investigated.

Experimentation shows that ARPRA is fairly robust for the specification of  $\psi_0$ . Even though a specification that fits an initial gamma distribution to IATs retrievable from the log shows the most accurate results, the observed performance differences are not statistically significant. Regarding  $n$ , ARPRA provides more accurate results for larger event logs than for smaller event logs. However, the

maximum event log size that is considered is 1600 cases, which might still be quite small in practice. Experimentation also showed that the effect of setting  $\tilde{n}$  equal to  $n$  is rather limited.

Future work on ARPRA will focus on the development of a more advanced parameter search strategy, taking into account the non-linear relationship between the gamma distribution parameters and  $q$ . Moreover, ARPRA should be extended to retrieve an IAT input model when (i) no a priori distribution is assumed and (ii) only start or end events are recorded in the event log.

## REFERENCES

- Blann, A. 2015. *Data Handling And Analysis*. 1st ed. New York : Oxford University Press.
- Demšar, J. 2006. “Statistical Comparisons Of Classifiers Over Multiple Data Sets.” *Journal of Machine Learning Research* 7:1–30.
- Henderson, S. G. 2003. “Input Modeling Uncertainty: Why Do We Care And What Should We Do About It?” In *Proceedings of the 2003 Winter Simulation Conference*, edited by S. Chick P. J. Sánchez, D. Ferrin, and D. J. Morrice, 90–100. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.
- Hillier, F. S., and G. J. Lieberman. 2007. *Introduction To Operations Research*. 9th ed. New York: McGraw-Hill.
- Kelton, W. D., R. P. Sadowski, and N. B. Zupick. 2015. *Simulation With Arena*. 6th ed. New York: McGraw-Hill.
- Law, A. M. 2007. *Simulation Modeling And Analysis*. 4th ed. New York: McGraw-Hill.
- Martin, N., B. Depaire, and A. Caris. 2014. “The Use Of Process Mining In A Business Process Simulation Context: Overview and Challenges.” In *Proceedings of the 2014 IEEE Symposium on Computational Intelligence and Data Mining*, 381–388.
- Martin, N., B. Depaire, and A. Caris. 2015. “Using Event Logs To Model Entity Arrival In Business Process Simulation.” Manuscript accepted for publication.
- Melão, N., and M. Pidd. 2003. “Use Of Business Process Simulation: A Survey Of Practitioners.” *Journal of the Operational Research Society* 54:2–10.
- Rozinat, A., R. S. Mans, M. Song, W. M. P. van der Aalst. 2009. “Discovering Simulation Models.” *Information Systems* 34:305–327.
- Song, M., and W. M. P. van der Aalst. 2007. “Supporting Process Mining By Showing Events At A Glance.” In *Proceedings of the 17th Annual Workshop on Information Technologies and Systems*, 139–145.
- van der Aalst, W. M. P. 2011. *Process Mining: Discovery, Conformance And Enhancement Of Business Processes*. 1st ed. Heidelberg: Springer-Verlag.

## AUTHOR BIOGRAPHIES

**NIELS MARTIN** is a PhD student at the Faculty of Business Economics of Hasselt University, Belgium. He is a member of the research group Business Informatics, where his research deals with the use of event log knowledge to construct more realistic BPS models. His e-mail address is niels.martin@uhasselt.be.

**BENOÎT DEPAIRE** is Assistant Professor at the Faculty of Business Economics, Hasselt University. He belongs to the research group Business Informatics. His research interests focus on the application of data mining, statistics and process mining within a business-related context.

**AN CARIS** is Assistant Professor of Operations Management and Logistics at Hasselt University (Belgium) within the Faculty of Business Economics. She takes a research interest in modelling intermodal freight transport networks, vehicle routing problems, simulation models and metaheuristics.