



Original Articles

An accelerated-time simulation of baggage traffic in an airport terminal

Q1 Gabriel Aguilera-Venegas, José L. Galán-García*, Enrique Mérida-Casermeiro,
Pedro Rodríguez-Cielos

Department of Applied Mathematics, University of Málaga, Spain

Received 10 October 2012; received in revised form 20 September 2013; accepted 27 December 2013

Abstract

This paper introduces the new ATISBAT model for generating accelerated-time simulations of baggage handling in an airport terminal. The philosophy of this model combines ideas from cellular automaton and neural network theories. The basics of this model was presented at ACA'2012 (in the Nonstandard Applications of Computer Algebra Session).

The algorithms involved in the ATISBAT model have been implemented using a Computer Algebra System since some symbolic computations are needed. On the other hand, a JAVA application that shows the simulation graphically has also been developed.

One of the most important characteristics of ATISBAT is its flexibility in two different ways. It allows for the simulation of a wide variety of baggage handling systems. It also allows users to choose from among a great list of random distributions (or even create an “ad hoc” one) to simulate the behavior of the random events within the system.

The accelerated-time simulations obtained with ATISBAT, can be used both as an aid during the terminal's design process (in order to optimize the baggage handling network's topology) and also to analyze the impact of any possible proposed changes in an already built baggage handling system.

The graphic interface developed produces important visual information about the simulation. This approach is very useful, since the effects of making any change to the system can be immediately visualized.

© 2014 Published by Elsevier B.V. on behalf of IMACS.

Keywords: Accelerated-time simulation; Baggage handling design; Baggage traffic simulation; CAS

1. Introduction

The handling of baggage is one of the most important activities in an airport. A good airport baggage handling system is crucial for both passengers and employees. A bad design may lead to undesirably long queues or even a collapse of the system.

The cost of a baggage control system is quite high. Therefore, it would be very important to test it in advance, using computer simulations, to determine the most appropriate system for an airport.

Q3 * Corresponding author. Tel.: +34 95 2 13 27 64; fax: +34 95 2 13 27 66.
E-mail address: jl.galan@uma.es (G. Aguilera-Venegas).

<http://dx.doi.org/10.1016/j.matcom.2013.12.009>

0378-4754/© 2014 Published by Elsevier B.V. on behalf of IMACS.

In this paper, we describe the ATISBAT (Accelerated Time Simulation of Baggage in an Airport Terminal) model in order to simulate the traffic of baggage in an airport. Other works which deal with accelerated-time simulation can be found in [2,8,14].

One of the most important characteristics of ATISBAT is its flexibility. ATISBAT is flexible both in a physical and a theoretical sense. In the physical sense, our model can simulate a great variety of baggage handling systems, not only for an specific airport but for many airports and different conditions. For example, the number of check-in desks, the number of output rings, the capacity of the warehouse, the number of safety levels and the length of the belts are all configurable in ATISBAT. In the theoretical sense, ATISBAT allows to choose among different standard random distributions for controlling how baggages access the system. Furthermore, the user can define an “ad hoc” random distribution for this matter. On the other hand, ATISBAT also allows one to choose among different standard random distributions or an “ad hoc” one, for controlling different characteristics of random phenomena occurring in the system. Some examples of these events are: accessing the warehouse, entering into a safety level zone, getting stuck on the conveyor belt or loosing an identification label.

ATISBAT combines ideas from cellular automaton methods and neural networks. In one hand, cells (pieces of the belt) have two possible states (empty or occupied) and the movement within the system is generated using cellular automaton theory. On the other hand, each occupied cell (suitcase) stores special characteristics in its state vector as in neural networks theory. The content of the state vector also affects the movement. That is, at any step, each cell changes its state depending on both the neighbor cells states and the individual values of the vector associated to the item in the cell (if occupied).

The idea of combining cellular automata methods and neural network theories to produce accelerated-time simulations has been previously explored by the authors in [2,10].

The first well known cellular automaton was Conway’s Game of Life [7]. This cellular automaton can be characterized as a simulation of the behavior of a community of cells. In each step of this simulation, cells die either because of overcrowding (more than three live neighbors) or because of under-population (fewer than two live neighbors). When a dead cell has exactly three live neighbors, a new live cell appears (reproduction). Otherwise, cells remain within their same state. These few rules produce movements in the system and simulate some kind of life.

Other cellular automata related to the simulation of movement are: Rule 184 [15], the Nagel-Schreckenberg model [12] and the Knospé-Santen-Schadschneider-Schreckenberg model [9].

ATISBAT has been implemented in two interconnected parts. The first, the multi-platform language JAVA, is used to develop the graphical user interface (GUI) for the system. The second uses a Computer Algebra System (CAS), specifically Maxima to deal with all mathematical computations within ATISBAT. The CAS is needed since some processes require symbolic computations.

This idea of developing a GUI for a CAS application, was also used in [2,3,11].

The basic aspects of ATISBAT model were introduced in the *Non-Standard Applications of Computer Algebra* Session of ACA 2012 Conference, held in Sofia.

In Sections 2 and 3, the physical aspects and mathematical model of ATISBAT, respectively, are described. Section 4 is devoted to the implementation of the ATISBAT model. Some results are shown in Section 5. Finally, the conclusions and future work are discussed in Section 6.

2. The physical level of ATISBAT

The physical level of ATISBAT is a flexible model that allows for the simulation of many baggage handling systems found in different airports. The general scheme and initial data have been obtained from the baggage handling system of the Málaga airport.

ATISBAT models the process followed by pieces of luggage from the check-in desks to the aircraft. The process from the arriving aircraft to the baggage claim is not considered since in most cases, this is an simple process which consists of carrying the pieces of luggage from the holds of the aircraft directly to the baggage claim area.

The physical level of ATISBAT model consist of several parts:

- The main baggage handling belt. It is made up of many individual conveyor belts forming the central ring (main ring) of the physical model.

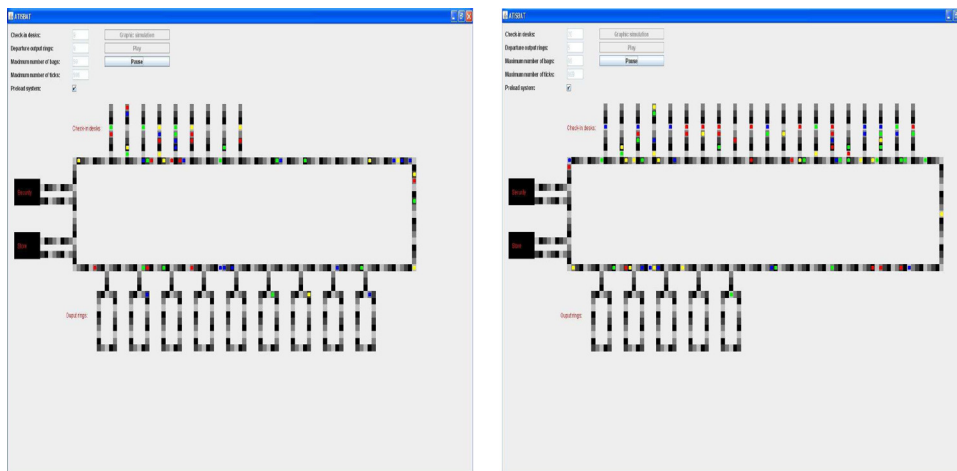


Fig. 1. The physical level of the ATISBAT model.

- The inputs of the physical model are the conveyor belts from the check-in desks. They transport the baggage labeled with its destination from the check-in desk to the main baggage handling belt.
- The outputs of the physical model are the output rings. People are in charge of carrying the baggage in these rings to transport it to the aircraft.
- The warehouse. It is a place where the pieces of baggage are stored when the time until the departure of its plane is long. They wait here until the departure time of the plane is approaching. Not all airports have this facility.
- Safety levels. Normally, there exist several safety levels. A suitcase is marked to enter the Level 1 safety zone when something wrong or suspicious is detected. For example a suitcase without identification label or a suitcase with a black box detected in the X-ray scanner. These suitcases are given more detailed examinations and either the problem is solved, for example a new label is stamped on the suitcase or the operator decides that the object is not dangerous. If this is not the case, the suitcase's safety level is increased. The controls the suitcase is required to pass are more sophisticated and specific in each upper safety levels. In the last safety level an expert (normally a policeman) decides if the suitcase can return to the system or must be given to the company because it cannot be transported into the plane or even it must be transported out of the airport in a safety place in order to remotely dispose of it.

All these physical parts are configurable in the ATISBAT model. Two graphical images of the physical model are shown in Fig. 1.

A piece of baggage starts the simulation in the ATISBAT model on the first position of the conveyor belt of a check-in desk. After this, the suitcase arrives to the main ring and it is then transferred to the output ring assigned to its flight. Along this way, different events can change this normal procedure and the suitcase could go through one (or more) of the following states:

- A suitcase lacks of an identification label, the label cannot be read properly or a suspicious object is detected in the X-ray scanner. If any of these situations are detected before the output of the first safety level, the suitcase is diverted to the safety zone 1. If it is detected after the safety output then the suitcase is forced to make a complete round in the main ring until it reaches the first level safety zone, where the problem will be analyzed.
- A suitcase stops suddenly in the conveyor belt without any known reason (for example, a jam). In this case a person is called to solve the problem. While the suitcase is stopped the rest of conveyor belts keep moving except when the next conveyor belt has a stopped suitcase. This situation could be compounded and could even provoke a complete collapse in the system.
- The departure time of the corresponding aircraft is too far in advance. In this case, the suitcase is driven to the warehouse (if this facility is available).

- The suitcase by passes its output ring (for example, the entrance is not empty). In this situation, in order to avoid a possible collapse, the suitcase will go around the main ring until it reaches its output ring again.

3. Mathematical model of ATISBAT

In this section, the mathematical model which ATISBAT uses to produce an accelerated-time simulation is described. It mainly consists of determining the movement dependent upon several situations and making decisions (generally, random ones) from among different possibilities.

There are different events for which the system will have to decide what suitcases have to do, such as: entering the system from check-in desks, moving on the conveyor belts, entering the warehouse, transferring to a safety zone, sudden stops, unexpected errors or exiting the system via their output ring.

In order to simulate such situations, the behavior of suitcases is described based upon random techniques which will allow ATISBAT to simulate the system, based upon different parameters. In each step, data is collected, thus obtaining information about possible improvements to the system design.

The need for using a CAS to implement ATISBAT is mainly due to these random simulations. In some cases, implicit computations are required.

3.1. Random sample generation

In this section we describe the library developed generating random samples for different continuous and discrete distributions. We also detail the use given to this library by ATISBAT.

The library developed is an adaptation from DERIVE to MAXIMA of the one previously developed by authors in [1]. This library allows the generation of samples for the most well-known distributions both, continuous and discrete.

Since any sample generation means working with random numbers, the first step was to implement an algorithm which improved the random functions available in MAXIMA and JAVA in order to obtain better samples. The algorithm chosen for this matter was `ran2` [13] which directly provides samples of a uniform distribution on the interval [0, 1]. With this starting point, the library contains different algorithms to generate samples from several continuous distributions (specifically: Uniform, Exponential, Normal, Lognormal, Weibul, Gamma, Beta, Chi-Square, Student's t , F , Z , Pareto, Logistic, Cauchy and Irwin-Hall distributions) and discrete distributions (specifically: Uniform discrete, Bernouille, Rademacher, Binomial, Poisson, Geometric, Negative Binomial and Hypergeometric distributions).

Furthermore, in order to make ATISBAT as flexible as possible, users are able to choose not only from the previous list of distributions, but can also specify an “ad hoc” distribution. This may be done for any of the different events that ATISBAT uses to control the system. Specifically, among the different techniques used to generate samples, the inverse transform method (both for continuous and discrete distributions) can be used to generate samples for an “ad hoc” distribution introduced by users. The use of this method for continuous distributions requires implicit integral computations, which can be only done in an exact form using a CAS.

3.2. ATISBAT and random distributions

In order to perform a simulation, ATISBAT considers different events that are generated using the random distribution generation library described in previous section. For each event, ATISBAT uses different distributions by default (although the user can change the distributions chosen). The way ATISBAT acts is as follows:

- **System entry.** Baggage enters the system through the different check-in desks. In order to simulate these entrances, ATISBAT uses the following two steps:
 - (1) The waiting time of next customer to be served. This is done, by default, using a random exponential distribution.
 - (2) The number of suitcases that the present customer wants to check-in. This is done, by default, using a random Poisson distribution.
- **Warehouse.** In order to simulate a suitcase entering the warehouse, a random continuous uniform is used by default. Once a suitcase enters the warehouse, a random exponential distribution will determine the waiting time in the warehouse before going back to the main ring.

- 154 • *Level 1 safety zone.* A suitcase is chosen to enter this first level safety zone using a random continuous uniform
155 distribution with a low value parameter as default. The waiting time is generated using, by default, an exponential
156 distribution.
- 157 • *Level 2 safety zone.* When a suitcase ends its waiting time in the first level, it can go back to the main ring or enter the
158 second level safety zone. This choice is done by means of a random continuous uniform distribution. Once again, the
159 waiting time in this second level of safety is computed using a random exponential distribution (both distributions
160 are used by default).
- 161 • *Level 3 safety zone.* ATISBAT acts as in the previous zone deciding randomly using a continuous uniform distribution
162 with a low value parameter whether the suitcase will go back to the main ring or move to the last safety level where
163 the suitcase is removed from the system. For example, in Málaga airport, the suitcase is carried to a special container
164 and is taken away from the airport and exploded (in the case of Málaga airport, this safety level has never reached
165 yet). The waiting time in this third level is also generated using a random exponential distribution (by default).
- 166 • *Output rings.* When a suitcase reaches its assigned output ring, it remains there until a person carries it to the airplane.
167 The waiting time within the output ring is computed by default using a random exponential distribution.
- 168 • *Sudden stop.* At every step and for all suitcases in the system, the probability for a specific suitcase stopping is
169 computed, by default, using a random continuous uniform distribution (with a very small value parameter). The time
170 this suitcase remains stopped is then computed using a random exponential distribution (by default). Note that the
171 rest of suitcases will keep moving within the ring unless the intended position is blocked.
- 172 • *Unexpected error.* If an unexpected error occurs (for example, when a suitcase loses its identification label), ATISBAT
173 controls it in a similar way as for a sudden stop, but in this case the suitcase is forced to go to the Level 1 safety
174 zone (this zone is assigned as the unique possible exit).

175 In any case, although the statistical distributions considered by default are the most common ones used in queueing
176 theory, the flexibility of ATISBAT allows the user to change any of the parameters and distributions with any other
177 parameters or distributions including an “ad hoc” one.

178 3.3. Movement

179 A suitcase enters the system throughout an active check-in desk. Let us remember that the number of check-in desks
180 and output rings are configurable.

181 A check-in desk can be active or inactive. The time in which a check-in desk becomes active and the time when it
182 moves to its inactive state are also configurable. By default, half of the check-in desks are set to active, but users can,
183 at any time, activate or deactivate any check-in desk.

184 The unit used for time within the system is call “step”. The conversion of steps to seconds, will mainly depend on
185 the velocity and length of the conveyor belts.

186 In the following, all parameters and random distributions are configurable although they all have some default
187 values.

188 Once a costumer is at an active check-in desk, the number of suitcases which are going to be entered onto the
189 conveyor belt associated with this check-in desk, is randomly generated using a Poisson distribution. The default
190 length of these conveyor belts has been set to 8 positions. In the first step, and if the random number of suitcases
191 is greater than zero, the suitcases are placed on the conveyor belt. A vector of states is then assigned to each of the
192 incoming suitcases. The state vector contains the number of its output ring and the different associated probabilities
193 (accessing the warehouse, accessing the different safety zones, sudden stop and system errors). In this first step, also
194 the number of waiting steps for next costumer in the check-in desk queue is randomly generated using an exponential
195 distribution.

196 As a general rule, in each step, the suitcases move one position forward, if possible. Otherwise, the suitcase will
197 remain in its present position.

198 Movement is checked from forward to backward and a suitcase will move to its next position if it is empty. A
199 suitcase in the main ring has priority over any suitcase that is not in it.

200 In order to describe the baggage movement, ATISBAT considers the following conflicting events which could break
201 the general rule:

- 202 • For all suitcases and for all steps, the probability of sudden stop is checked. If the decision is to stop, the suitcase will
203 remain in its present position the number of steps determined by the corresponding random exponential distribution.
- 204 • For all suitcases and for all steps, the probability of system error is checked. If an error occurs, the probability of
205 entering the Level 1 safety zone is set to 1 and the suitcase is redirected to this zone.
- 206 • If two suitcases should be assigned to move to the same position, the one on the main ring will have priority. This
207 event can occur in the following situations: a suitcase wants to access the system from a check-in desk, from the
208 warehouse or from the safety zone. Also, if a suitcase leaves the main ring to its exit ring, this suitcase will have
209 priority over a suitcase which is circling in this exit ring.
- 210 • If a suitcase reaches a free entrance (at the security zone or the warehouse), it will have two options: keep moving
211 in the main ring or entering the appropriate zone. This decision is randomly chosen, by ATISBAT, depending on the
212 probabilities associated to the suitcase for accessing the zones.
- 213 • If a suitcase reaches its output ring exit and it is free, it will move to it.
- 214 • If a suitcase reaches an occupied entrance (at the security zone, the warehouse or its output ring), the suitcase will
215 check the next position in the main ring. If it is empty, the suitcase will move to it or, if it is occupied, it will remain
216 in its present position.
- 217 • The state of a suitcase in the queue of its output ring is checked before the states of the suitcase in the ring. That is, if
218 there is any conflict, a suitcase in the output ring will have to give way to an incoming suitcase from the main ring.

219 Finally, a random exponential distribution is associated by default to each output ring in order to determine the
220 number of steps after which the suitcases in the output ring are removed (simulating that they are carried to the holds
221 of the aircrafts).

222 4. The GUI and its connection with the CAS

223 ATISBAT has been implemented combining the graphical facilities of Java and the mathematical engine behind
224 a CAS, specifically Maxima. The work carried out in JAVA provides a flexible GUI for the ATISBAT method whose
225 mathematical core has been developed in MAXIMA. Most of the mathematic computations made in MAXIMA could
226 have been done in JAVA. However, some operations need symbolic computation (such as implicit integration). This fact
227 caused us to decide on using a CAS which deals with exact and symbolic computations.

228 The interconnection between JAVA and a CAS has been explored by the authors in previous works [2,3]. More
229 specifically, the combination of JAVA with MAXIMA was also described in [11]. The work consisted of the development
230 of a GUI for the free numerical software SCILAB using MAXIMA when symbolic calculation was required. This GUI
231 provided an environment similar to the commercial software MATLAB.

232 Although the authors are experts in the CAS DERIVE [4–6], we chose MAXIMA for several reasons:

- 233 • DERIVE was discontinued in 2007. Although it will still run on all present Windows operating systems, it is no longer
234 supported.
- 235 • MAXIMA is an open source software.
- 236 • We find programming in MAXIMA to be much more user-friendly than programming in DERIVE.
- 237 • The communication with JAVA can be done in background mode with MAXIMA but it is not possible with DERIVE.
- 238 • From a didactical point of view, we used to work with DERIVE in Mathematic lectures for Engineering degrees. But
239 since DERIVE was discontinued we decided to migrate to MAXIMA, the CAS our students were used to working with
240 in other subjects.

241 Nonetheless, the mathematical core of ATISBAT can be implemented in any other CAS software which allows
242 programming and communicating with JAVA.

243 Regarding the graphical approach of ATISBAT, its GUI was implemented in JAVA mainly due to:

- 244 • It allows developing multi-platform pieces of software because of JAVA virtual machines.
- 245 • Its graphical capabilities satisfied our particular needs.
- 246 • Communication between processes is supported.

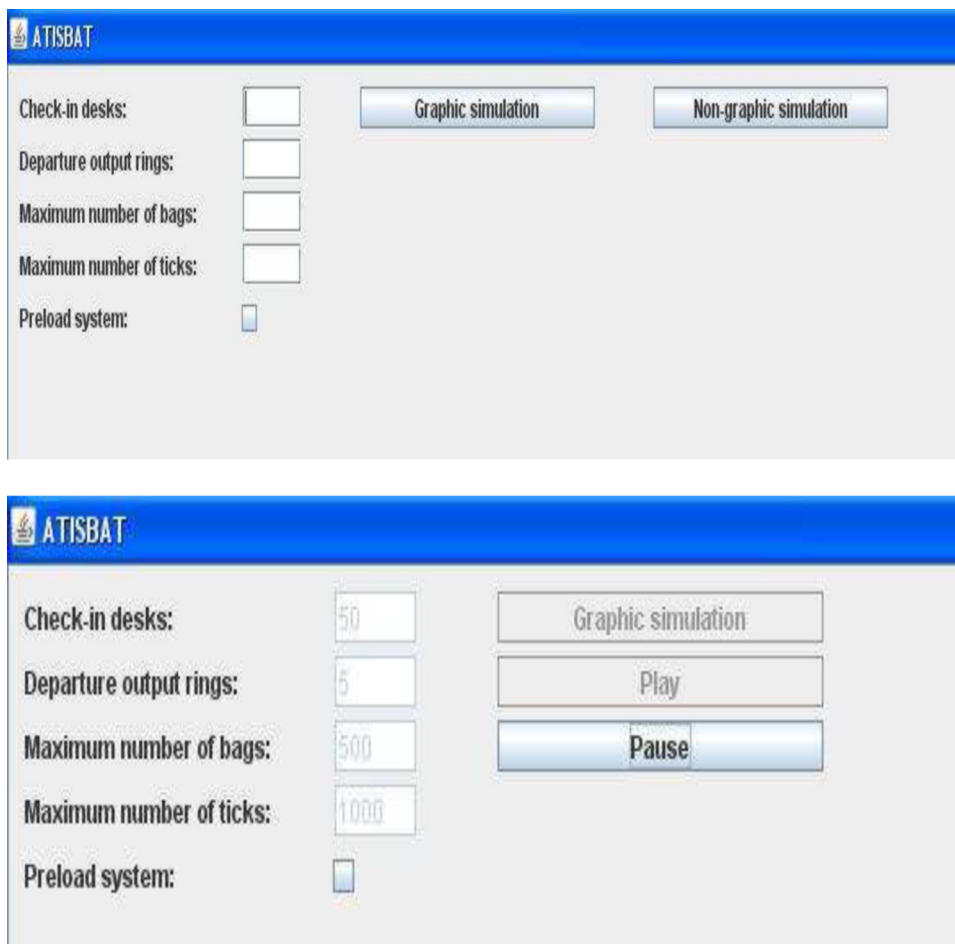


Fig. 2. Graphical interface for simulations.

- The authors had previously worked in this environment.
- It is a widely used object oriented programming language.

The GUI developed for the ATISBAT method allows for displaying each step of the accelerated-time simulation or just the final result. Also, the GUI also allows the user to enter different options (parameters) in order to configure the simulation. In Fig. 2, two pictures of the graphical interface are shown.

The implementation of the ATISBAT model (both MAXIMA and JAVA modules) can be downloaded from <http://www.matap.uma.es/jlgalan/ATISBAT/>.

5. Results

As an example of accelerated-time simulations using ATISBAT, a stress test to the system has been carried out. A series of simulations increasing the number of active check-in desks and fixing the other parameters have been made. Specifically, the fixed parameters are:

- Distribution function for the waiting time to the next customer in each check-in desk: Exponential (mean 50).
- Probability function for the number of suitcases checked by the customer: Poisson (mean 2).
- Output rings: 5.
- Probability function for the output ring assigned to each suitcase: Discrete Uniform between 1 and 5.

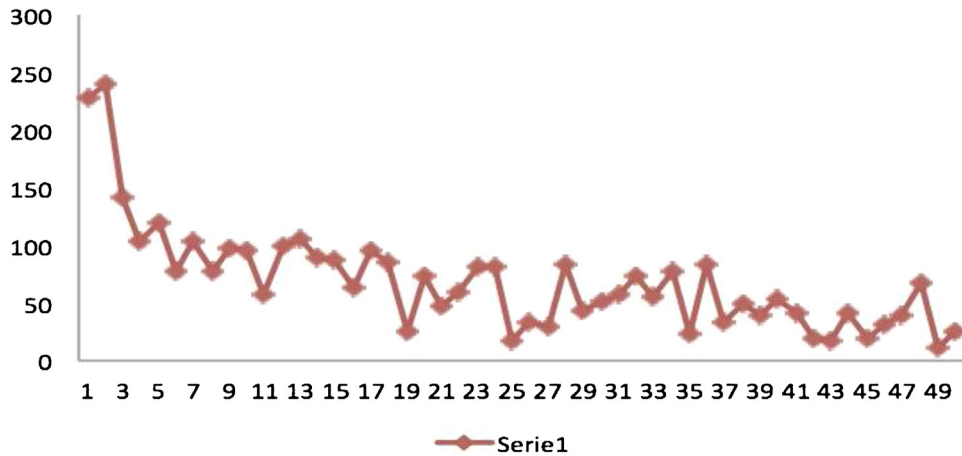


Fig. 3. X: Number of active check-in desks. Y: Steps to the first problem.

- Distribution function for sudden stop of each suitcase: Uniform in $[0.01, 0.001]$ and the number of steps the suitcase is stopped is computed using an Exponential (mean 50).
- Distribution function for error of each suitcase: Uniform in $[0.01, 0.001]$.
- Distribution function for the warehousing of each suitcase: Uniform in $[0.01, 0.001]$. The steps the suitcase is stopped is computed using an Exponential (mean 200).
- Distribution function for accessing the Level 1 safety zone: Uniform in $[0.01, 0.001]$. The number of steps the suitcase is stopped is computed using an Exponential (mean 100).
- Distribution function for accessing the Level 2 safety zone from Level 1: Uniform in $[0.01, 0.001]$. The number of steps the suitcase is stopped is computed using an Exponential (mean 100).
- Distribution function for accessing the Level 3 safety zone from Level 2: Uniform in $[0.01, 0.001]$. The number of steps the suitcase is stopped is computed using an Exponential (mean 100).
- Distribution function for accessing the Level 4 safety zone from Level 3: Uniform in $[0.001, 0.0001]$.
- Distribution function for the waiting time to remove the suitcases in each output ring: Exponential (mean 200).
- Length of the conveyor belt from each check-in desk to the main ring: 8 positions.
- Length of the conveyor belt from main ring to the warehouse and the way back: 8 positions.
- Length of the conveyor belt from main ring to the safety zones and the way back: 8 positions.
- Length of the conveyor belt from main ring to the output rings: 4 positions.

The results of these accelerated-time simulations are drawn in Fig. 3.

As expected, the graphic shows that the bigger number of active check-in desks, the smaller number of steps needed to reach the first problem. This is due mainly to the combination of these two facts:

- If the number of active check-in desks is high, the number of suitcases entering the system increases quickly and therefore, it is more likely that there exists a stopped suitcase. Furthermore, the possible stopped suitcase(s) may provoke others to stop and block the output of some check-in desk, provoking a problem in its conveyor belt.
- Since the number of output rings is fixed to 5 in this example, the percentage of suitcases leaving the system is similar for low or great number of check-in desks. This means that, when the number of check-in desks is high, the flux of suitcases entering the system is greater than those leaving, resulting in queues and a great number of suitcases in the main ring. Therefore, this fact may cause jams in the main ring and also in the different conveyor belts of check-in desks, warehouse and safety zones.

6. Conclusions and future work

The ATISBAT method allows generating accelerated-time simulations, adapted to the user necessities, for baggage traffic control in an airport. This fact provides a useful tool in order to decide the design of the baggage handling system.

The great flexibility of ATISBAT allows adapting the simulation to a wide variety of baggage handling systems. This flexibility provides not only accelerated-time simulations for new systems but also simulations for any possible change in an already built system.

Nowadays, smart traffic lights and smart traffic signals are being considered as a possibility of improvement of traffic in cities. In fact there are several international projects which deal with this idea (for example in Pittsburgh and Málaga). One direction for future work related to this idea will be to adapt the philosophy of the ATISBAT model to develop accelerated-time simulations of car traffic using smart traffic lights and signals. Since this new technology is still in development and making physical tests is expensive, accelerated-time simulations can be presented as an easy and cheap way to simulate the impact of these smart devices in different cities. Furthermore, for a given city, different designs can be simulated and decide which one is the best option.

Another related work will consist on developing accelerated-time simulations for city bus networks. This kind of simulations could be used both for developing new designs or for improving already existing ones.

Acknowledgements

We thank our friend Carl Leinbach for editing the English language of the paper. His comments and suggestions as a mathematician and an English native speaking person have improved the paper. We also thank the anonymous reviewers for their valuable comments and corrections provoking also an improvement of the paper. Any remaining error must be attributed to the authors.

References

- [1] G. Aguilera, J.L. Galán, M.A. Galán, Y. Padilla, P. Rodríguez, Generating random samples from continues and discrete distributions with DERIVE, *DERIVE News*, 75 (2009) 22–43.
- [2] G. Aguilera, J.L. Galán, J.M. García, E. Mérida, P. Rodríguez, An accelerated-time simulation of car traffic on a motorway using a CAS, *J. Math. Comput. Simulat.* (2012), <http://dx.doi.org/10.1016/j.matcom.2012.03.010>.
- [3] G. Aguilera, J.L. Galán, R. Madrid, A.M. Martínez, Y. Padilla, P. Rodríguez, Automated generation of contrapuntal musical compositions using probabilistic logic in DERIVE, *J. Math. Comput. Simulat.* 80 (6) (2010) 1200–1211.
- [4] J.L. Galán, From Vienna to Tartu: a 10 years tour using DERIVE in TIME, in: Proceedings of the International Conference TIME 2012, Tartu, Estonia, 2012.
- [5] J.L. Galán, P. Rodríguez, M.A. Galán, Y. Padilla, Multiple integration.dfw: Line, double, triple and surface integrals, User Contributed Math Packages of software DERIVE 6. Texas Instruments, 2004.
- [6] J.L. Galán, P. Rodríguez, M.A. Galán, Y. Padilla, E. Sánchez, J. Sánchez, Complex analysis.dfw: Complex numbers and functions, User Contributed Math Packages of software DERIVE 6. Texas Instruments, 2004.
- [7] M. Gardner, The fantastic combinations of John Conway's new solitaire game "life", *Sci. Am.* 223 (1970) 120–123.
- [8] A. Hernando, E. Roanes-Lozano, A. García-Álvarez, An accelerated-time microscopic simulation of a dedicated freight double-track railway line, *Math. Comput. Model.* 51 (2010) 1160–1169.
- [9] W. Knospe, L. Santen, A. Schadschneider, M. Schreckenberg, Towards a realistic microscopic description of highway traffic, *J. Phys. A: Math. Gen.* 33 (48) (2000) 477–485.
- [10] E. Mérida, G. Aguilera, J.L. Galán, P. Rodríguez, A new neural model for traffic simulation, *Lect. Notes Comput. Sci.* 6254 (2010) 471–476.
- [11] A. Mora, J.L. Galán, G. Aguilera, A. Fernández, E. Mérida, P. Rodríguez, Scilab and maxima environment: towards free software in numerical analysis, *Int. J. Technol. Math. Educ.* 17 (2) (2010) 75–80.
- [12] K. Nagel, M. Schreckenberg, A cellular automaton model for freeway traffic, *J. Phys.* 2 (12) (1992) 2221–2229.
- [13] W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery, Numerical Recipes in C. The Art of Scientific Computing, Cambridge University Press, United States, 1999.
- [14] E. Roanes-Lozano, L.M. Laita, E. Roanes-Macías, An accelerated-time simulation of departing passengers' flow in airport terminals, *Math. Comput. Simulat.* 67 (2004) 163–172.
- [15] S. Wolfram, *A New Class of Science*, Wolfram Media, 2002.