International Journal for Quality Research 13(4) 823–836
ISSN 1800-6450

Marcelo F.N. Henriques [1]
António A.C. Vieira
Luís M.S, Dias
Guilherme A.B. Pereira
José A. Oliveira

Article info:
Received 11.06.2019
Accepted 05.09.2019

UDC – 519.876.2
DOI – 10.24874/IJQR13.04-05

ANALYSIS OF AN ELEVATOR SYSTEM USING DISCRETE EVENT SIMULATION: CASE STUDY

Abstract: *This paper documents the work conducted to simulate an elevator system, using SIMIO software. The modelled system represents a case study that was analyzed in a hospital at Braga, Portugal. A previ-ous work on the same case study concluded that the best dwell time configuration would be around 10 seconds, however it did not consider the impact of different client demand on the elevator system. In this sense, this paper analyses the impact of both parameters on the performance of the system. This will be achieved by analyzing the impact on the total time spent by clients in the system, the number of clients inside the system, and waiting for the elevator, waiting time, average elevator occupation and number of elevator movements. Conclusions and future work agenda were discussed in the conclusions section.*

Keywords: *Elevator; Management systems; Agents Modelling; 3D Simulation; SIMIO; Case study*.

1. Introduction

Moving people and cargo in a vertical way is the most typical objective of an elevator system. In the core of an elevator system there is its management system, responsible for deciding the next elevator movement to be performed, through its algorithm, based on multiple inputs. A basic algorithm for a sys-tem with only one elevator installed, as is the model used in the present case study, can be described as follows (Setchi, 2010):

- Move in a certain direction, up or down, stopping at all floors, where there are calls or destina-tions registered;
- Change its direction, when there are no calls or destinations at floors beyond the current floor in the current direction, or when it reaches the last floor, changing from going down to going up –when it reaches the bottom floor – or changing from going up to going down – when it reach-es the upper floor;
- Stop, in case there are no calls or destinations registered in the system.

Changing an elevator system, or simply its algorithm, can be costly, and can imply system inopera-bility for a period of time. By simulating an elevator system, it is possible to change the system partially or entirely in a virtual way, allowing for a measurement of such changes in the system performance, without stopping the current system or invest in a trial and error approach. This tactic allows for cost savings and avoid system inoperability. These measurements of the changes in the system performance allow top management to take decisions based on simulation data. The simulation approach can help both top management when deciding which system to implement, and elevator OEMs when defining system parameters.

[1] Corresponding author: Marcelo F. N. Henriques
Email: marcelo.henriques@dps.uminho.pt

Examples of such parameters are, for example, elevator capacity or dwell time; the latter representing the time that the elevator remains at each floor, with its doors open, to let clients in and out.

Dwell time is a parameter crucial to all system KPIs (Key Performance Indicators), but mainly for the total time of each client (waiting time plus travel time). A high dwell time may:

- Increase the probability of clients entering the elevator at the floor it stopped, thus diminishing client waiting time on that floor;
- Increase the waiting time of clients calling the elevator at other floors.
- Increase travel time, due to the increment of stop time at each floor where the elevator stops in between his or her origin and destination floor.

By contrast, a low elevator dwell time may:

- Decrease the probability of clients entering the elevator at each floor in which the elevator stops, as the “opportunity window” is smaller, and, consequently, increase the waiting time of such clients;
- Decreasing the waiting time on other floors where clients are waiting, due to the following point;
- Increase the elevator movements, because it spends less time stopped;
- Increase the energy consumed by the system, as there are more movements, due to more mo-tor starts to move the elevator cabin and its counter-weight.

A balance among these points is needed, making the dwell time value essential for clients, to whom a fast arrival to the destinations is important; and system owners, that strive for a good system perfor-mance on both time and energy. Thus, a balance between energy and time is needed.

This paper uses the elevator model demonstrated in a previous work (Henriques et al., 2016), where an elevator system was modelled using the SIMIO software to evaluate the recommended dwell time of a given client demand. In this paper, different client demand scenarios, called intensity scenarios, were created. To this end, the simulation model was adapted, in order to add the necessary properties to model the intensity scenarios, which replicate a different client demand.

This paper starts with the present introduction, to highlight the importance of this study. The section number 2, Literature review, describes a literature review of elevator systems and discrete simulation, focused on the tool used: the SIMIO software. Section 3, Simulation Model, demonstrates the elevator model developed and the process responsible for the elevator algorithm. The following section, number 5, Simulation Experiments, details the parameters defined for the experiments ran and the data retrieved on six graphs.

2. Literature review

Most recent models of elevator group management systems (e.g. Destination Dispatch) had, in their genesis, tests and data retrieved from using computing simulation. One simulation tool that outstands in the elevator industry is the software Elevate® (Barney & Al-Sharif, 2015), which allows to simulate and analyse elevator traffic, with support for different configurations and applications, e.g. two floors eleva-tors, an elevator system with different speeds and different attending floors (“About Elevate”, 2016). This software runs on Windows™ and was developed by the London-based company Peters Research. Another innovation by this company is the software Elevate Live™, which allows checking the status of the elevator management system in real time (“About Elevate Live”, 2016).

This software is not the only simulation tool used in the elevator industry, but it is one of the most referred and promoted. But, considering the unwill to share information, protect intellectual property and maintain a market advantage, companies of this industry tend to not reveal which tools are used.

But the need and use of simulation in this field is real (Barney & Al-Sharif, 2015; Hakonen & Siiko-nen, 2009; Zhang & Zong, 2014), because elevator models can reach high levels of complexity. Taking for instance the Shangai Tower, where hundreds of elevators travel vertically, with certain restrictions and different purposes, the level of complexity associated to this system becomes obvious.

The number of simulation tools can be very high. Thus, its comparison becomes a very important task. However, most scientific works related to this subject "analyse only a small set of tools and usually evaluating several parameters separately, to avoid making a final judgement due to the subjective na-ture of such task" (Dias et al., 2007).

Hlupic and Paul (1999) compared a set of simulation tools, distinguishing between users of software for educational purpose and users in industry. In his turn, Hlupic (2000) developed "a survey of academ-ic and industrial users on the use of simulation software, which was carried out in order to discover how the users are satisfied with the simulation software they use and how this software could be further im-proved". Dias et al. (2007) and Pereira et al. (2011) comparing a set of tools based on popularity on the internet, scientific publications, WSC (Winter Simulation Conference), social networks and other sources claim: "Popularity should never be used alone otherwise new tools, better than existing ones would never get market place, and this is a generic risk, not a simulation particularity" (Dias et al., 2007), however, a positive correlation may exist between popularity and quality, since the best tools have a greater chance of being more popular. According to the authors, the most popular tool is ARENA, (Kelton et al., 2009), and the good classification of SIMIO is noteworthy. Based on these results, Vieira et al. (2014) compared both tools taking into consideration several factors. This last referenced paper is also a good source of information for researcher and practitioners, since it compares SIMIO with the most popular tool (ARENA), giving some basic examples.

SIMIO has two main levels for modelling. One more simple called 'Facility', suitable for practition-ers without computer science background, where one can create models in a building-block approach over a physical layout, providing a realistic 3D animation. The second level, called 'Process', enables the creation of detailed behaviour using logical flow charts to specify virtually anything.

Processes, once created, can be used anywhere in the 'Facility' level. Moreover, processes can be "attached" to Entities (objects), enabling them to react actively and autonomously. This behaviour pushes SIMIO "living" objects to agents. It is controversial to consider SIMIO objects as intelligent, once such term has a connotation to support logical programming and self-learning ability.

Another relevant capability is the support for object class hierarchy, allowing the extension of exist-ing objects rather than creating from scratch, i.e. an object can be generic and used multiple times inside another object, that can be part of a model, e.g. a house can be an object inside a neighbourhood, that is an object of a city model.

SIMIO is based on intelligent objects (Sturrock & Pegden, 2010; Pegden, 2007; Pegden & Sturrock, 2011). These "are built by modellers and then may be used in multiple modelling projects. Objects can be stored in libraries and easily shared" (Pegden, 2013). Unlike other object-oriented systems, in SIMIO there is no need to write any programming code, since the process of creating a new object is completely graphic (Pegden & Sturrock, 2011; Pegden, 2007; Sturrock & Pegden, 2010). The activity of building an object in SIMIO is identical to the activity of building a model. In fact, there is no difference between an object and a model (Pegden, 2007; Pegden & Sturrock, 2011). A vehicle, a customer or any other agent of a system are examples of possible objects and, combining several of these, one can represent the components of the system in analysis. Thus, a SIMIO model looks like the real system (Pegden & Stur-rock, 2011; Pegden, 2007). This can be very useful, particularly while presenting the results to someone unfamiliar to simulation.

In SIMIO, the model logic and animation are built in a single step (Pegden & Sturrock, 2011; Pegden, 2007). This makes the modulation process very intuitive (Pegden & Sturrock, 2011). Moreover, the animation can also be useful to reflect the changing state of the object (Pegden, 2007). In addition to the usual 2D animation, SIMIO also supports 3D animation as a natural part of the modelling process (Sturrock & Pegden, 2010). To switch between them the user only needs to press a specific key (Sturrock & Pegden, 2010). Moreover, SIMIO provides a direct link to Google Warehouse (Pegden & Sturrock, 2011).

SIMIO offers two basic modes for executing models: interactive and experimental. In the first it is possible to watch the animated model, which is useful for building and validating the model. In the sec-ond, it is possible to define properties of the model that can be changed (Sturrock & Pegden, 2010).

3. Simulation model

In this section, the developed simulation model will be covered. A detailed description of it can be found on a previous publication (Henriques et al., 2016).

In this context, the developed process responsible for enhancing the elevator object and transforming it into an agent is the one depicted in Figure 1. This process is initiated upon the start of each model run, and is executed on an infinite loop.

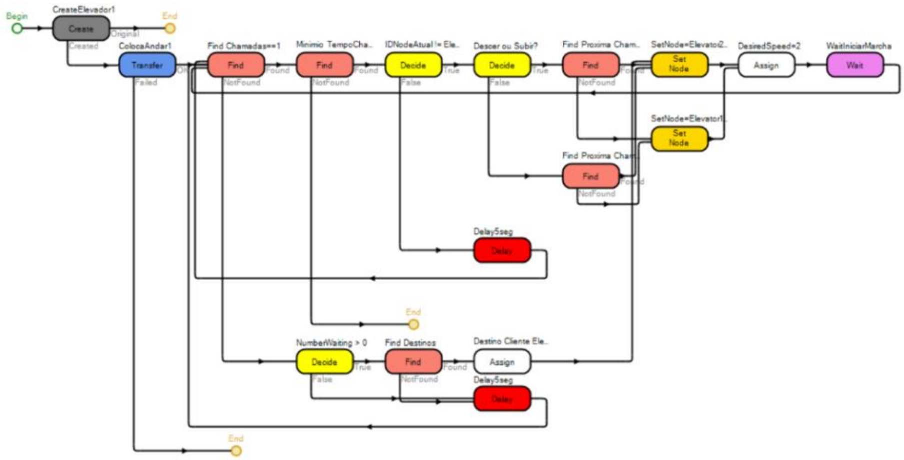


Figure 1. Process containing the elevator algorithm

This process is responsible for creating the elevator and giving it the intended behaviour, such as travelling inside the elevator shaft, wait for clients, decide which floor to go next, among other aspects. This further configure an agent modelling approach, since the tool allows the user to detail the behaviour of all objects, including entities, as is the case of the clients and the elevator itself.

To ensure the elevator stops at all floors which have calls registered or a client inside the elevator wants to exit, the process represented in Figure 2 is executed whenever the elevator arrives at a given node that represents a floor.

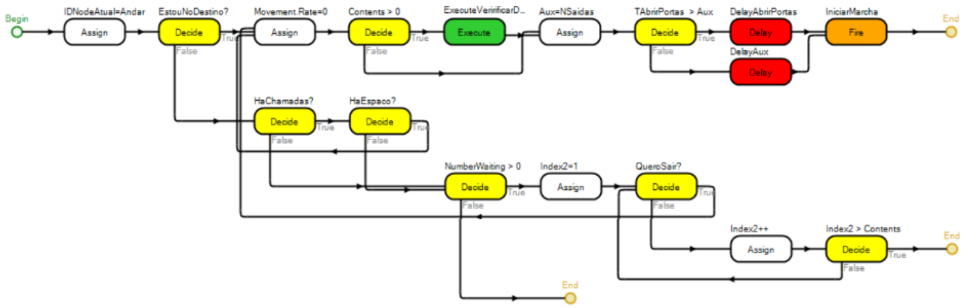


Figure 2. Process responsible for deciding an elevator stop

This process analyses if the elevator has arrived on the floor which was assigned to it as a destina-tion in the previous process. If not, it will be verified if there are calls placed on that floor, ensuring that the elevator still has room to let clients in, or if any client riding it wants to exit at the current floor. If the current node is the elevator destination, has a call placed or is the destination of a client inside the eleva-tor; the next steps will ensure that the elevator stops, and will model the dwell time, allowing clients to exit or to enter onto the elevator. Afterwards, an event will be fired to indicate that the elevator can resume its trip, allowing the process represented in Figure 1 to continue its loop. In this regard, communication be-tween these two processes is necessary and ensured, since both processes are executed in parallel.

To raise clients from objects to agents, the process illustrated in Figure 3 is run upon client creation.

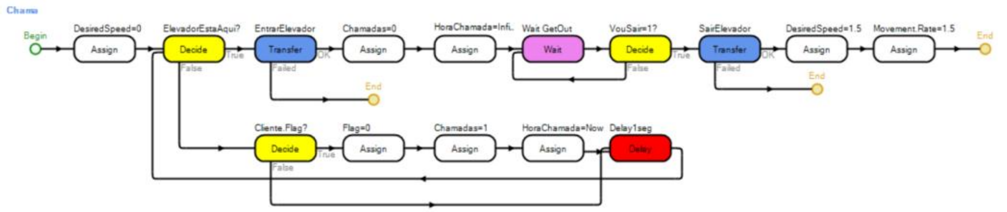


Figure 3. Process ran by clients while inside the system

This process is responsible for assuring each client waits for the elevator, calls the elevator, enters it when the elevator is on same floor as the client, and the client gets out of the elevator onto the last path into the sink of the destination floor. Figure 4 shows the model during its run.

As can be seen, the model is comprised by seven floors. As they are created, entities travel until the end of the respective paths of each floor, where they will be able to call the elevator, through another process. Thereafter, the elevator evaluates the many requests it has received and travels to the floors to allow the entrance and exit of clients. Finally, clients are transported to the intended destination and eliminated of the system.

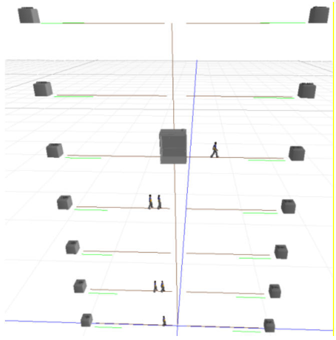


Figure 4. Simulation model during its execution

In the next section, simulation experiments conducted are analysed.

4. Simulation experiments

Experiments are one of the most important tools in SIMIO, as it allows testing the impact of different model properties, in which each combination of values corresponds to a different scenario. The properties of the model used are:

- Random exponential value for inter-arrival time of the source placed on the ground floor (number one): regulating the inter-arrival time between each client creation on that floor;
- Random exponential value for inter-arrival time of the sources placed on the upper floors (number two to seven): regulating the inter-arrival time between each client creation on those floors;
- Dwell time: time in which the elevator remains at each floor, with its doors open, to let clients in and out;
- Elevator capacity: maximum number of clients that the elevator can transport.

In this paper, the elevator capacity will be linear at 21 clients. This is a high number in order to not being a limitation in high demand cases, i.e. in cases where a high value of clients is created, this elevator capacity will be in line with systems installed on high demand applications, not being the system bottleneck.

The dwell time will be changed from 1 to 20 seconds, in order to have a good representation of this property.

The creation of clients will range from an average of 473,0 to 1296,6 clients per hour, which is a high demand on the elevator model, which has only one elevator. This is achieved with the inter-arrival time on a random exponential of 0,8 and 0,9, to 0,1 to 0,2 minutes, on the ground floor and upper floors, respectively, on both cases. Table 1 gives a better representation of client creation:

Table 1. Value of the properties in the Source objects and clients generated per hour

Intensity Acronym	SIMIO Expression		Average number of entities created per hour
	Ground floor	Upper floor	
i1	$Random.Exponencial(0{,}1)$	$Random.Exponencial(0{,}2)$	1032,54
i2	$Random.Exponencial(0{,}2)$	$Random.Exponencial(0{,}3)$	1016,99
i3	$Random.Exponencial(0{,}3)$	$Random.Exponencial(0{,}4)$	923,03
i4	$Random.Exponencial(0{,}4)$	$Random.Exponencial(0{,}5)$	818,17
i5	$Random.Exponencial(0{,}5)$	$Random.Exponencial(0{,}6)$	710,04
i6	$Random.Exponencial(0{,}6)$	$Random.Exponencial(0{,}7)$	612,85
i7	$Random.Exponencial(0{,}7)$	$Random.Exponencial(0{,}8)$	763,74
i8	$Random.Exponencial(0{,}8)$	$Random.Exponencial(0{,}9)$	474,87

Table 1 shows the SIMIO expression that was able to model the different intensities of creation of entity clients. It is important to note that these expressions regulate the inter-arrival time, in minutes, where a lower inter-arrival time translates into a bigger generation of entity clients.

The left column shows the 'Intensity Acronym' – the term used on the remaining of this document to refer to each of the intensity scenarios. The centre columns show the expression used in SIMIO to model the different inter-arrival time on each scenario. The centre left values correspond to the ground floor, while the centre right values are for the upper floors. The expression considered for the upper floors is slightly higher – translating to a higher inter-arrival time, thus creating less entity clients – as the ground floor is the main entrance one, thus being the most used. Lastly, the right column shows the average number of entities created per hour in the system. These values represent a mixed movement of clients inside the system, replicating a typical week day during work hours.

A warm-up time of 1 hour was used. This decision was made due to the fast reach of a warm-up state by the model, as the creation of clients is of a high frequency.

To decide the number of replications tests were conducted. A total of 25 tests were made, keeping all properties with the same value: 10 seconds for dwell time, 21 clients for the elevator capacity and 0,5 and 0,6 for the random exponential in the inter-arrival time, respectively, in minutes. The number of replications was changed from 1 to 25. Table 2 shows the change in data as the number of experiments increases. Each value represents the average of the data retrieved in each scenario.

Table 2. Impact of replications on the KPIs

Replications	Average total time [minutes]	Average waiting clients	Average occupation	Average number of clients in the system	Average elevator movements	Average waiting time [seconds]
1	2,59	21,22	10,30	31,88	6899,00	23,04
2	3,03	26,57	10,33	37,26	6901,00	23,02
3	3,61	33,61	10,34	44,30	6902,00	23,00
4	3,40	31,02	10,34	41,71	6899,25	22,96
5	3,32	30,25	10,33	40,92	6897,60	22,95
6	3,26	29,51	10,32	40,18	6898,33	22,96
7	3,13	27,90	10,33	38,58	6897,71	22,97
8	3,09	27,49	10,34	38,17	6898,13	22,98
9	3,02	26,56	10,33	37,23	6897,89	22,98
10	2,98	26,13	10,32	36,80	6897,80	22,97
11	2,93	25,51	10,33	36,19	6897,64	22,95
12	2,93	25,62	10,34	36,31	6897,67	22,95
13	2,88	24,92	10,34	35,61	6897,38	22,94
14	2,84	24,42	10,34	35,10	6896,57	22,93
15	2,81	24,10	10,34	34,79	6896,87	22,93
16	2,79	23,86	10,33	34,54	6896,56	22,93
17	2,79	23,83	10,33	34,50	6896,59	22,93
18	2,79	23,80	10,33	34,47	6896,33	22,92
19	2,84	24,45	10,33	35,13	6896,79	22,93
20	2,81	24,09	10,33	34,77	6896,50	22,92
21	2,79	23,91	10,33	34,58	6896,86	22,92
22	2,80	24,01	10,32	34,68	6896,91	22,93
23	2,79	23,87	10,32	34,54	6896,74	22,93
24	2,79	23,84	10,32	34,51	6896,75	22,93
25	2,77	23,59	10,33	34,27	6897,12	22,93

By evaluating these values, it is possible to conclude that the numbers tend to stabilize around 16 to 20 replications, depending on the parameter analysed. As such, the number of replications chosen was 20.

Once the model was developed and validated, data was retrieved from it, in order to get relevant information that would lead to conclusions about the developed model. One of the major benefits of using SIMIO is the possibility of conducting simulation experiments on a model.

A simulation experiment allows for executing a set of scenarios with different values for the model properties, and the impact of those changes on the model KPIs (Key Performance Indicators). In the present model, dwell time is the main model property in study, being changed from 1 to 20 seconds. The other model properties changed on each scenario was the inter-arrival time of the floor sources – mentioned above and enumerated on **Table *1***. The other implemented property is the capacity of the elevator, that can be analysed in future studies.

In total, 160 scenarios were run (1 to 20 seconds of dwell time, times i1 to i8 intensities), with 20 replications each. To conduct the experiments a laptop computer was used, with the following specifications: Intel Core i7-3630QM processor, clocked at 2,40GHz, and 16GB of DDR3 RAM. The processing time of each replication was between 6 to 140 seconds, depending on the intensity scenario.

The dwell time is crucial to the total time of a client (waiting time plus travel time) because if it is increased, it increases the probability of clients entering the elevator at a floor, thus diminishing client waiting time on the current floor; but will also increase the waiting time of clients in other floors. If this time is decreased, the probability of clients entering the elevator at each stop decreases and the elevator will move more, thus decreasing the waiting time on other floors. A balance between these two possibilities needs to be found. In order to have a good representation of the impact of this property on all KPIs, the value will vary from 1 to 20 seconds. To note that a value of dwell time with a good performance on a specific KPI, e.g. average client total time, at up-peak time can have a bad performance on a mixed or down-peak movement of clients, as calls can be placed in a more focused area of the building, e.g. the ground floor, or can be spread across all floors. The focus was, therefore, to analyse the impact of dwell time in the system performance, namely the following KPI established:

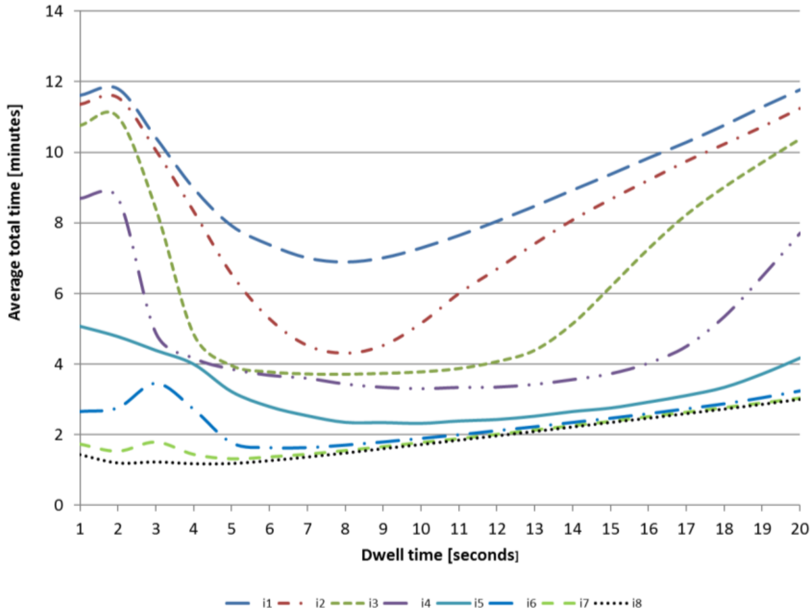
- Average total time in the system, per client: sum of waiting time and travel time, of the clients;
- Average elevator occupation (or load): number of clients riding the elevator;
- Elevator movements: number of movements executed by the elevator in the simulation runtime;
- Average number of clients waiting: number of clients that are waiting for the elevator, in all floors;
- Average waiting time per client.

In order to ensure that the results do not contain irrelevant data, as a result of the time needed for the system to achieve a "full-operating status", it is very important to define an accurate warm-up period. In this context, a warm-up period of 3600 seconds was defined because, on the several tests conducted, it was found that from this time on, the KPI values achieved a more stable status. Furthermore, 10 replications were used, to ensure that different random number seeds are used. The simulation time in the experiments was 24 hours. Graphs 1 to 6 illustrate the obtained results. The first graph (Graph 1) shows a relation between the dwell time of the elevator and the average total time in minutes spent by clients in the system.

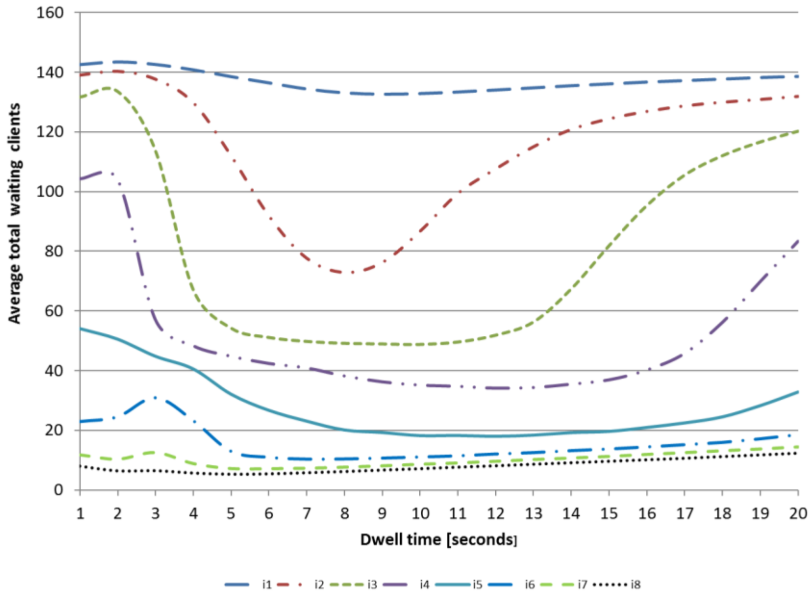
According to Graph 1, the scenarios modelled with higher intensities were more sensitive to the change of the dwell time, than the scenarios modelled with lower intensities. Furthermore, it can also be stressed that the dwell time with the best values for the scenarios with lower intensity are around 5 to 7 seconds, for i6, i7 and i8. Scenarios i5 and i4 have the best performance at around 9 to 11 seconds. Sce-narios i3, i2 and i1 achieve the lowest total time at around 7 to 9 seconds.

These values are explained through the opportunity window of dwell time, when the client is arriving and the elevator is still in the floor, and he or she is able to enter it. This graph and the analysis show the importance of a good evaluation of the site where the elevator system will be installed, i.e. the dwell time to be implemented will depend on the client demand.

Graph 2 shows the relation between the dwell time of the elevator and the average number of clients waiting for the elevator.



Graph 1. Average total time per client in minutes

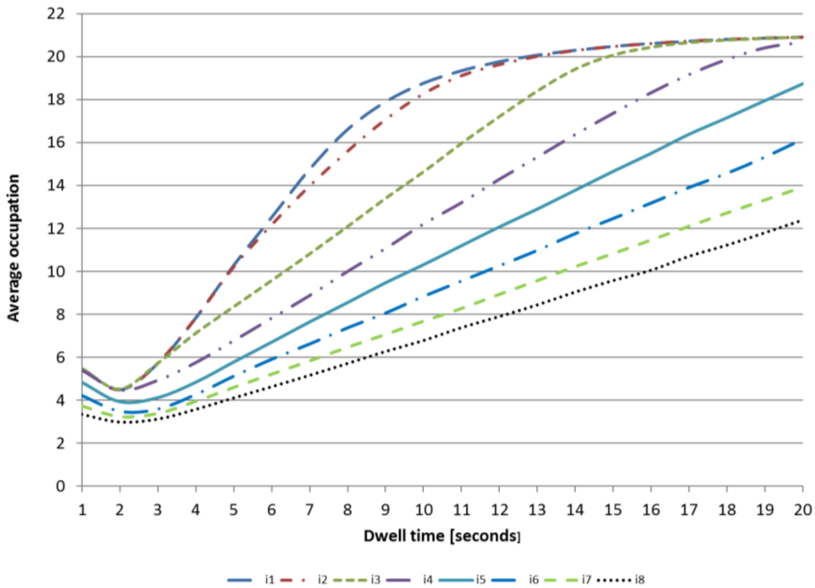


Graph 2. Average total waiting clients

By analyzing Graph 2, it can be observed that for high intensities, as well for lower intensities, the KPI in question suffers slight changes, whilst for medium intensities, the variation of the KPI, in response to different dwell times of the elevator is higher. This low response is explained in two stages: in the lower intensities, the elevator is ready to answer the calls coming from the arriving clients, as there are few peo-ple in the system, and there are not many clients waiting. While in the higher intensities the elevator sys-tem is already in too much stress, not being able to answer as quickly to each call that comes in, i.e. the elevator stops at almost all floors, leaving and receiving clients.

In the i6, i7 and i8 lower intensities, the lowest values are reached at the 5 to 8 seconds band. The i4 and i5 scenarios achieve the lower average total waiting clients at 11 to 14 seconds. The higher intensities i3, i2 and i1 have the lowest average at around 9 to 11 seconds.

Graph 3 illustrates the change in the average elevator occupation, as the dwell time increases, for the eight different client creation scenarios.

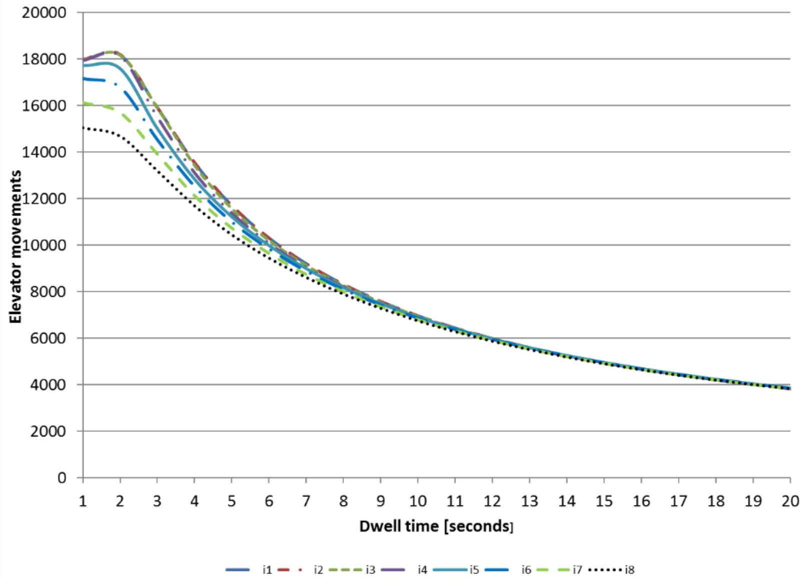


Graph 3. Average occupation of the elevator

The average occupation in Graph 3 shows that the high intensity scenarios reach the maximum oc-cupation of the elevator faster, at 21 clients. At the 17 seconds of dwell time, i1, i2 and i3 scenarios reach the average occupation at the maximum capacity of the elevator. The i4 scenario reaches it at 20 sec-onds. This demonstrates that these scenarios put the system into its limits.

As for dwell time performance, this KPI shows that dwell time will affect the occupation in a propor-tional way, except for the 1 second time, i.e. after the 2 seconds time, the average occupation grows pro-portionally to the increase of the dwell time property. This is explained by the less time available to enter the elevator: if the dwell time is low, less clients are able to enter the elevator and ride it.

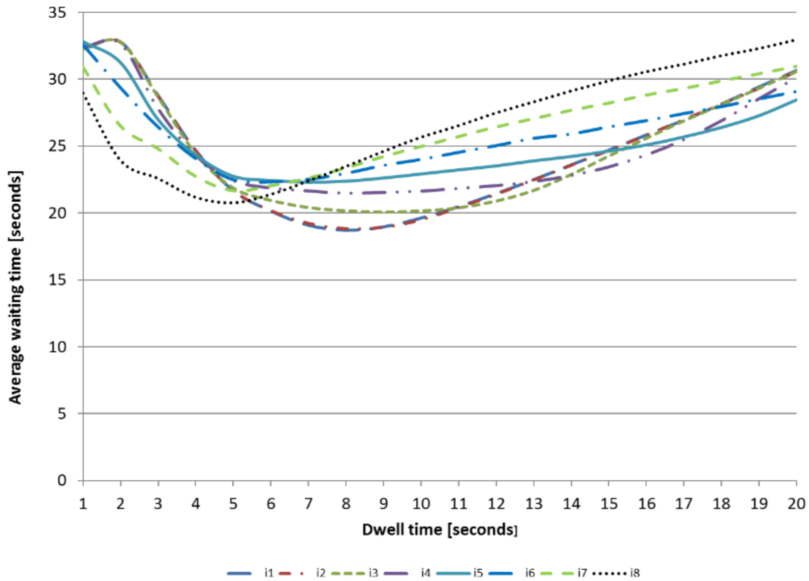
Graph 4 represents the impact of dwell time on elevator movements, in the different scenarios.



Graph 4. Number of movements executed by the elevator

Graph 4 demonstrates the negative impact of a high dwell time on the number of elevator move-ments. With a high dwell time, despite having a higher intensity of clients, the elevator will move no more than with a lower intensity scenario. It is confirmed that a low dwell time will increase the number of elevator movements, thus increasing the energy consumption of the system.

Graph 5 shows the average waiting time of clients in the system as a response to dwell time, for the various intensity scenarios.

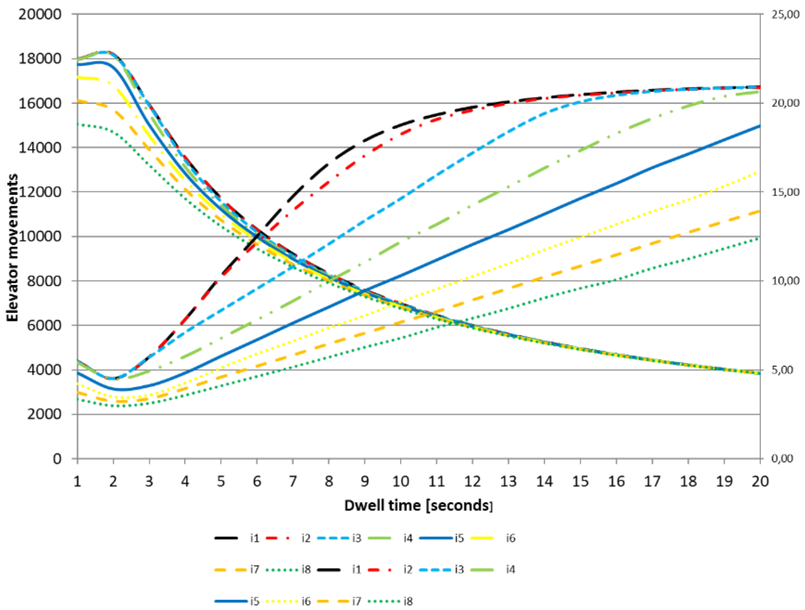


Graph 5. Average waiting time per client in seconds

Graph 5 shows that the minimum values of average waiting time, for all intensity scenarios, are reached at around 5 to 8 seconds of dwell time. It is important to stress that these values are low, at 18 to 23 seconds of waiting time.

The scenarios order in the graph registers a change at around 4 seconds. While, towards 15 and 20 seconds, some scenarios, especially the higher ones, change its direction. This reorder at around 4 seconds of dwell time demonstrates the more sensitive nature of lower intensity scenarios to dwell time, i.e. as there is a low intensity of clients, a high dwell time does not benefit those systems, as the elevator is stopped for longer periods, but no more clients enter it beside the one that placed the call.

Graph 6 confronts the average occupation with the number of elevator movements. These two KPIs are the basis for the system energy consumption. The left vertical scale shows the number of elevator movements, representing the decreasing lines along the graph. The average elevator occupation is shown in the increasing lines along the graph, having the right vertical scale as its reference.



Graph 6. Relation between elevator movements and average occupation in function of dwell time

Graph 6 shows that the average occupation is the key KPI when evaluating the energy consumption of the system, on a comparison of different scenarios, as the number of elevator movements in all the scenarios tends to the same value, independently of the intensity of each scenario.

Upon evaluating the average elevator occupation, it is possible to see that, as the intensity decreas-es, a high dwell time will correspond to the point of less energy consumption, as the 40% elevator capaci-ty – usually the value calculated for the counter-weight – mark of the lower intensity scenarios is at the right side of the graph.

5. Conclusions

An **elevator** system was modelled in **SIMIO** - a recently developed discrete simulation tool. The simulation model was based on an Hospital located in the north of Portugal. The tool was chosen due to its similarities to ARENA - the most used simulation tool worldwide - since they were developed by the same authors. Moreover, it fully supports 3D animation, which results on very appealing simulation models, which also contributes for a better understanding of the system in its execution.

To evaluate the performance of the system, the following Key Performance Indicators (**KPI**) were defined: average total time; average occupation; number of elevator movements; average of waiting clients on all floors; and average waiting time.

By analyzing the graphs, it is possible to conclude that the lower intensities i8, i7 and i6 reach the best performance at around **5 to 7 seconds**, consuming less energy at the higher dwell time value. The middle intensities in scenarios i5 and i4 have the best performance at around 11 seconds. The higher intensities i3, i2 and i1 reach the best performance at around **9 seconds** – confirming the conclusions from the previous study. These values show that there is not a direct relation between dwell time and client demand, i.e. the lower and high intensity scenarios benefit from a lower dwell time, as there are few clients to attend or many other calls in other floors to attend, respectively; whilst on the medium intensity scenarios, the bigger dwell time improves the system performance.

These values enhance the importance of retrieving data from the site in which the elevator system will be implemented, in order to have a good idea of the client demand and adapt the system to it. A good performance can only be achieved if the elevator system is adapted to its environment.

By re-using previously defined SIMIO objects in other models, this elevator model can be used on **future research**. For instance, in a multiple elevator system, where ETD (Estimated Time to Destination), or other algorithms, could be implemented. Furthermore, the power consumption of the elevator system could also be quantified, as well as the consideration to implement different management systems, taking advantage of the modelling of the elevator as an entity; therefore, giving more options to the modeller, as the entity logic is already in place, and its properties were defined in a way that the user can simply modify its values in a simple combo box.

Acknowledgments: This work has been supported by FCT – Fundação para a Ciência e Tecnologia in the scope of the project: PEst-OE/EEI/UI0319/2014 (ALGORITMI).

References:

About Elevate. (n.d.). Retrieved January 23, 2016, from https://www.peters-research.com/index.php/elevate/elevate-elevate-express

About Elevate Live. (n.d.). Retrieved January 23, 2016, from https://www.peters-research.com/index.php/elevate-live/more-about-elevate-live

Barney, G., & Al-Sharif, L. (2015). E*levator Traffic Handbook: Theory and Practice* (2nd Edition). Routledge.

Dias, L., Pereira, G., & Rodrigues, G. (2007). A Shortlist of the Most Popular Discrete Simulation Tools. *Simulation News Europe*, *17*, 33-36

Hakonen, H., & Siikonen, M.-L. (2009). *Elevator Traffic Simulation Procedure*. Lift Report.

Henriques, M., Vieira, A., Dias, L., Pereira, G., & Oliveira, J. A. (2016). Impact of Dwell Time on Vertical Transportation through Discrete Simulation in SIMIO. In A. Paias, M. Ruthmair, S. Voss (Eds.), *7th International Conference on Computational Logistics*, *2016* (pp. 411-426). Springer International Publishing Switzerland.

Hlupic, V., & Paul, R. (1999). Guidelines for selection of manufacturing simulation software. *IIE Transactions*, *31*, 21-29.

Hlupic, V. (2000). Simulation software: an Operational Research Society survey of academic and industrial users. In J. A. Joines, R. R. Barton, K. Kang, P. A. Fishwick (Eds.), Proceedings of the 2000 Winter Simulation Conference (pp. 1676-1683). Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

Kelton, W. D., Sadowski, R., & Zupick, N. (2009). *Simulation with Arena* (5th Edition). McGraw-Hill Educa-tion.

Pegden, C. D. (2007). SIMIO: A new simulation system based on intelligent objects. Simulation Conference, 2007 Winter, 9-12 Dec. 2007. (pp. 2293-2300).

Pegden, C. D., & Sturrock, D. T. (2011). Introduction to SIMIO. In *Proceedings - Winter Simulation Conference*, *2011*, Phoenix, AZ, pp. 29-38.

Pegden, C. D. (2013). Intelligent objects: the future of simulation. Retrieved from http://www.simio.com/resources/white-papers/Intelligen-objects/

Pereira, G. B., Dias, L.S., Vik, P., & Oliveira, J. (2011). Discrete Simulation Tools Ranking – a Commercial Software Packages comparison based on popularity. *ISC 2011 – 9th Industrial Simulation Conference*, June 6-8, 2011, Venice, Italy, pp. 5-11.

Setchi, R. (2010). Knowledge-Based and Intelligent Information and Engineering Systems. In R. Setchi, I. Jordanov (Eds.), *Knowledge-Based and Intelligent Information and Engineering Systems*, *1*, 133-134. Cardiff, UK: Springer Science & Business Media.

Sturrock, D. T., & Pegden, C. D. (2010). Recent innovations in SIMIO. *Proceedings - Winter Simulation Conference*, *2010*, Baltimore, MD. pp. 21-31.

Vieira, A., Dias, L. S., Pereira, G. B., & Oliveira, J. A. (2014). Comparison of SIMIO and ARENA Simulation Tools. In *ISC'2014, 12th Annual Industrial Simulation Conference*. EUROSIS, Skovde, Sweden, Jun 11-13. pp. 5-13.

Zhang, J., & Zong, Q. (2014). Energy-saving-oriented group-elevator dispatching strategy for multi-traffic pat-terns. *Building Services Engineering Research and Technology*, *35*(5), 543-568. http://doi.org/10.1177/0143624414526723

Marcelo F.N. Henriques
University of Minho,
Campus Gualtar, 4710-057,
Braga, Portugal
marcelo.henriques@dps.uminho.pt

António A.C. Vieira
University of Minho,
Campus Gualtar, 4710-057,
Braga, Portugal
antonio.vieira@dps.uminho.pt

Luís M.S, Dias
University of Minho,
Campus Gualtar, 4710-057, Braga, Portugal
lsd@dps.uminho.pt

Guilherme A.B. Pereira
University of Minho,
Campus Gualtar, 4710-057,
Braga, Portugal
gui@dps.uminho.pt

José A. Oliveira
University of Minho,
Campus Gualtar, 4710-057,
Braga, Portugal
zan@dps.uminho.pt