ANALYZING THE IMPACT OF QUANTUM SIZE ON THE ACCURACY AND PERFORMANCE OF CELL-DEVS FIRE MODELS

Ala'a Al-Habashna
Cristina Ruiz-Martin
Gabriel Wainer

Department of Systems and Computer Engineering
Carleton University
1125 Colonel By Dr.
Ottawa, ON, CA
{alaaalhabashna, cristinaruizmartin, gwainer}@sce.carleton.ca

ABSTRACT

Fire spreading is a phenomenon that has received much interest over the years. Because the analysis of fire spreading using mathematical models can be extremely challenging due to the various factors involved, researchers use modeling and simulation to study this problem. There are many fire spread models. However, many of them require long execution time, which reduces their usability. Quantization was proposed as a solution to deal with the long execution time problem. The primary goal of this research is to study the effect of quantum size on the accuracy and performance of Cell-DEVS fire-spread models, and to develop an improved version of existing Cell-DEVS fire-spread models.

Keywords: Fire, Quantization, Cell-DEVS, Modelling, Simulation.

1 INTRODUCTION

Fire spreading is a complex phenomenon that received the attention of governments and the population. Every year, there are reports of many fire incidents all over the world and it is a constant threat to forests and wildlife. For example, "wildland fires" have been destroying an average of 2.5 million hectares a year in Canada alone, since 1990 (NRC 2019). Fires are also common in cities where they have been occurring increasingly. For example, the Ottawa Fire Services (OFS) responded to 25,048 incidents in 2017 (4.4% more than in 2016) (Pingitore 2017). Firefighters need tools that help them to decide where to put the efforts to control and extinguish the fires. This decision is not straightforward because the evolution of the fire depends on a lot of variables such geography, wind, composition of the soil, etc.

Although Cellular Automata have been used to define fire models (Muzy et al. 2004; Balbi, Santoni, and Dupuy 1999), this approach has some precision constrains and takes extra computation time. These disadvantages can be overcome with the use of Cell-DEVS (Wainer and Giambiasi 2002; Wainer 2009). Using Cell-DEVS, a cell space is described as a discrete event model in which explicit delays can be used to model accurately the cell timing properties.

An example of a fire model with Cell-DEVS is the one by (Muzy et al. 2005). The main issue of this model is that it takes long simulation times, which reduces its usability in real life scenarios, where firefighters need timely decisions. In (MacLeod, Chreyh, and Wainer 2006), the authors propose an improved version of the above mentioned model to decrease the simulation time by reducing the number of messages exchanged in the model. In the next section, we discuss in detail a new version of this model,

and build on top of it to provide an improved version that not only reduce execution time, but also provide more accurate results.

We analyze the effect of quantum size in the model proposed by (MacLeod, Chreyh, and Wainer 2006) on performance and accuracy. Then, we propose an alternative implementation of the model using the advanced features of CD++ (ports and multiple state variables) to eliminate auxiliary cells to further improve performance and accuracy. When we decrease the number of cells in the model, the number of messages exchanged is also reduced.

The structure of the paper is as follows. In section 2, we present the background of this research. We explain Cell-DEVS, Q-DEVS, and the base model used in this research. In section 3, we discuss the effect of quantum size on the performance of the model. In section 4, we present an alternative model and implementation of the model taking advantage of the advanced characteristics of CD++ to eliminate auxiliary cells of the model in the second and third planes. Finally, section 5 presents the conclusions of this research.

2 BACKGROUND

Fire spreading is a complex phenomenon that many researchers have been working on over many decades. It is a complicated topic to analyze because the spread of fire depends on many different variables such as the burning materials, the weather, the geography of the affected area, etc. As such, modeling the phenomena of fire spread analytically is very difficult. Various mathematical models for fire spread have been developed. Some of these models are purely theoretical (Pastor et al. 2003; Fons 1946). Such models are based on existing characteristics of burned materials and mathematical models such as material conduction characteristics, heat transfer equations, etc. An example of such models is the one in (Fons 1946) where energy conservation equation were used to model fire spread versus logarithmic growth of fuel bed temperature. Different models based on this theoretical approach have also been developed by other researchers such as the models in (Hwang and Xie 1984; Albini 1985).

Other mathematical models for fire spread incorporate empirical results that can be obtained from real-life experiments in addition to theoretical foundations. Popular models that have been developed using this approach are (Rothermel 1972; McArthur 1966). However, such models usually depend on the conditions of the experiments performed, and always need calibration to be employed in different environments. For further information on mathematical fire spread models, the reader is referred to (Pastor et al. 2003).

The models above provide a foundation for understanding and analyzing fire behavior. However, there is a need for spatial models and tools that can be employed during the firefighting process. A popular approach for this purpose is computational fluid dynamics (CFD) modeling of fire. With this approach, the collective behavior of fire spread in an area is studied as a continuum. For examples of models based on this approach, the reader is referred to (McGrattan et al. 2012; Novozhilov 2001). Although such approaches are extremely useful in studying and analyzing fire spread, entity-based spatial models that take into account the characteristics of each subarea should be used, as they are more accurate.

Cellular Automata-based fire spread models have been developed and used to study fire spread (Muzy et al. 2004; Balbi, Santoni, and Dupuy 1999). However, such models require a relatively high computation time and produce less accurate results. These disadvantages can be overcome with the use of Cell-DEVS (Wainer and Giambiasi 2002; Wainer 2009). Using Cell-DEVS, a cell space is described as a discrete event model in which explicit delays can be used to model accurately the cell timing properties. An example of such models is the fire-spread model in (Muzy et al. 2005). In (MacLeod, Chreyh, and Wainer 2006), the authors propose an improved version of the above mentioned model to decrease the simulation time by reducing the number of messages exchanged in the model. The authors in (MacLeod, Chreyh, and Wainer 2006) propose to combine the model with Quantized-DEVS (Q-DEVS) (Wainer and Zeigler 2000) and “dead reckoning”. Both techniques are based on the transformation of a continuous signal (i.e. the temperature over time) in a piecewise polynomial function, where the important event is not the actual temperature but when the temperature falls into a new interval. The authors explain the rules of the new version of the model. However, they do not differentiate between the Q-DEVS and the “dead reckoning”

approximation in their model definition. Their new version of the model uses three planes: the main one containing the fire propagation and two auxiliary ones.

In this work, we present an empirical study of the effect of quantization size on the accuracy of the model, which is a very important aspect. Furthermore, we build on top of the model in (MacLeod, Chreyh, and Wainer 2006) to provide an improved version that does not only reduce execution time, but also provide more accurate results. In the following, we introduce the two methodologies we are using in this research; Cell-DEVS and Q-DEVS. We also present two existing fire spread Cell-DEVS models: the model in (Muzy et al. 2005) and its quantized version (MacLeod, Chreyh, and Wainer 2006).

2.1 Cell-DEVS and Q-DEVS

Discrete Event System Specification (DEVS) (Zeigler, Praehofer, and Kim 2000) is a formalism for modeling complex Discrete Events Systems based on the Set Theory and Systems Theory. DEVS has a hierarchical and modular structure that allows defining models independently and couple them to work together. The connection is done by input and output ports and exchanging messages between the models. The resulting model can be coupled to other model defining a hierarchical structure of multiple layers.

In DEVS, there are two types of models atomic and coupled. Atomic models define the behavior of the system, and coupled models describe the hierarchical structure. One of the advantages of the hierarchical and modular structure of DEVS is that it allows reusing models and reduces development and testing times. Moreover, model definition, implementation, and simulation are separated (i.e. we can implement the same model in different DEVS simulators such as such as JDEVS (Filippi and Bisgambiglia 2004), DEVSJava (Sarjoughian and Zeigler 2014), CDBoost (Vicino et al. 2015), CD++ (Wainer 2009), etc.

Cell-DEVS formalism is an extension to DEVS to implement cellular models with timing delays (Wainer and Giambiasi 2002; Wainer 2009). In Cell-DEVS, models are defined as a lattice of cells. Every cell is defined as an atomic DEVS model, and the cell space is as the coupled model. Each cell can communicate with input and output messages with its neighbors, but also with other DEVS models outside the cell space. The neighborhood can be uniform (i.e. the same for all cells) or non-uniform (i.e. different cells can have different neighbors). Moreover, the neighbor cells can be adjacent or include remote cells. Each cell also has a state that is updated using a local transition function and a set of external events.

In this research we use the CD++ simulator in (López and Wainer 2004), an extended version of CD++ (Wainer 2009) that allows using input and output ports and multiple state variables in a cell. A local transition function is defined as a set of rules using the following grammar:

rule: POSTCONDITION DELAY { PRECONDITION }

When the PRECONDITION is satisfied, the state of the cell will change to the designated POSTCONDITION. The new values of the state variables of the cell will be transmitted to other components after consuming the DELAY using different ports. If the precondition is false, the next rule in the list is evaluated until a rule is satisfied or there are no more rules. In order to combine DEVS with continuous components, the Q-DEVS discretization showed potential when combined with Cell-DEVS (Wainer and Zeigler 2000). The main idea is to represent continuous functions or signals in the cells by the crossing of an equal spaced set of boundaries. Instead of periodically checking what the value of the state is at each time, we focus on when a state variable enters in a new given state.

2.2 The original fire-spread model and its quantized version

The fire model in (Muzy et al. 2005) uses Cell-DEVS and is based on the mathematical fire-spread model in (Rothermel 1972). According to this model, the temperature curve is divided into four stages, and at any given instant, each cell in the model will be in one of these stages. The first is the inactive stage, where a cell has no neighbors with a temperature higher than the ambient temperature (Ta). The second is the unburned stage in which the temperature of the cell is increasing due to heat from the neighboring

cells; during this stage, the cell's temperature is between the ambient temperature and the ignition temperature (300 ºC). The third is the burning stage in which the cell has reached the ignition temperature and fuel in the cell starts to burn. During this stage the cell's temperature increases until it reaches a peak temperature, and then falls back down to 60 ºC. A part of the burning section of the temperature curve is shown in Figure 1. The temperature keeps decreasing until reaching 60 ºC, at which point, the fire reaches its fourth stage (i.e., burned).

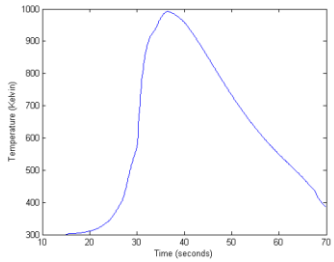

Figure 1: Simplified Temperature Curve.

In the original model, the area under study is divided into a mesh of cells. Each one represents a small region of the studied area. Furthermore, the Cell-DEVS definition contains two planes to store the states for each region. The first plane represents the fire spread itself and stores the temperature of each cell. The additional plane is used to store the ignition times of the cells, i.e., records the simulation time when the corresponding cell in the fire spread plane reaches the ignition temperature. The neighborhood of the model is shown in Figure 2 and the rules to update the fire spread plane are as follows:

- **Unburned phase**: the cell temperature is calculated as the weighted average of the current cell's temperature and its neighbors' temperatures.
- **Burning phase:** the temperature is again calculated as the weighted average of the temperatures around it, but with an additional factor, that considers burning section of the temperature curve.
- **Inactive and burned phases**: the cell's temperature does not change and thus these cells stay in the passive state.

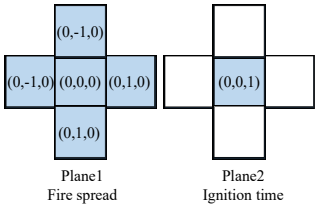

Figure 2: Cellular model Neighborhood.

In (MacLeod, Chreyh, and Wainer 2006), the authors tested this model. To increase efficiency and usability of the model, a quantized version was defined including the following changes:

- Unburned cells are passive until the ignition temperature is reached. By analyzing the model's equations, it was found that a cell in the unburned phase would reach the ignition temperature when one of its neighbors reaches a temperature above 650 °K or when two of its neighbors reach a temperature above 474 °K. As such, an unburned cell can be kept passive until one of these events occurs. This reduces the number of active cells, and consequently reduces the number of messages exchanged between cells. Furthermore, this also reduces the number of computations executed.

- Quantized Cell-DEVS was used to reduce the number of calculations and exchanged message, which significantly reduces execution time. With Quantized Cell-DEVS, all cells in the model have a fixed quantum size and each cell has a quantizer. The idea is that each cell will only send output to its neighbors if its temperature has passed into the next quantum threshold. The quantizer acts as the detector

that discerns when a threshold is crossed, and sends out the output to the neighboring cells. This way, an output is sent only if the threshold has been crossed. Q-DEVS can be thought of as increasing the amplitude quantum size. Using Q-DEVS the number of messages exchanged between cells can be reduced. However, the state (value) of cells still need to be checked at the same frequency, which means that the number of computations will not be decreased considerably. This can be improved when combined with dead reckoning. Messages will be sent to the neighboring cells only if the temperature reaches certain points. Furthermore, the time needed to reach a certain temperature is calculated, and the cell is passivated until that time. i.e., time is calculated as a function of temperature rather than calculating temperature as a function of time. In addition to reducing the number of messages transmitted, this also reduces the number of calculations, which significantly reduces execution time.

To find the function that gives the time as a function of temperature in the burning stage, the inverse of the temperature curve (Figure 1) was calculated for the increasing and decreasing parts of the curve. Then, the time to reach a certain temperature, T_2, can be calculated as the difference $f(T_2)$- $f(T_1)$, where $f(T_1)$ and $f(T_2)$ give the time to reach temperatures T_1 and T_2, respectively.

In the quantized model, the cells' temperatures remain on the first plane and ignition times are moved to the third plane. However, it contains one additional plane over the original model that helps to determine which rule we should apply to update the cells' temperature. This is called the "supplementary info" plane. Table 1 lists the states of the cells in the second plane.

Table 1: States used in the second plane.

State value	**Description (all temperature values are in °K)**
0	Initial and default value
-100	Cell burning but its temperature is not high enough to cause a neighbor to ignite ($301 \leq T < 474$)
-200	Cell burning, and its temperature is high enough to cause a neighbor to ignite, if another neighbor is in the same state ($474 \leq T < 650$)
-300	Cell burning, and temperature high enough to cause a neighbor to ignite by itself ($650 \leq T < 992$)
-400	Temperature of the cell reached the peak value (992), temperature is decreasing
-500	Cell is burnt out

The neighborhood of the model's first plane (the fire spread plane), which is depicted in Figure 3, contains the Von Neumann neighborhood in the same plane, as well as the corresponding cell from the second (ignition times) and third (supporting information) planes.

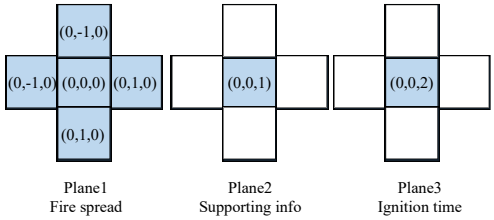

Figure 3: Neighborhood of the quantized fire-spread model.

The fire spread rules of the model for each cell are as follows:

- **Burning up phase**: this includes cells that are burning but have not yet reached the peak temperature (the corresponding state in plane 2 is -100, -200, or -300). The delay is calculated according to the inverse of the increasing part of the burning section of the curve in Figure 1.

- **Burning down phase**: cells that reached their peak temperature and are still burning, their temperature is decreasing but they have not yet reached the burnt-out stage. The state in plane 2 is -400. For these cells, the delay is calculated according to the inverse of the decreasing part of the burning section of the curve in Figure 1.
- **A cell whose state in plane 2 is 0 and has a temperature between 301 and 474**: the cell has not ignited. The state in plane 2 will be -100 after a short time, i.e., the cell will ignite but its temperature is below 474 °K.
- **A cell whose state in plane 2 is 0 or -100, and has a temperature greater than or equal to 474:** the cell's state in plane 2 will change to -200 after a short delay. This indicates that the cell has ignited and has reached 474 °K.
- **A cell whose state in plane 2 is 0 or -200, and has a temperature greater than or equal 650**: the cell's state in plane 2 will change to -300 after a short delay. This indicates that the cell has ignited and has reached 650 °K.
- **A cell whose state in plane 2 is 0 or -300, and has a temperature greater than or equal 992**: the cell's state in plane 2 will change to -400 after a short delay. This indicates that the cell has reached the peak temperature, and that the burn down part of the curve should be used
- **A cell whose state in plane 2 is -400, and its temperature is less than or equal 333**: the cell's state in plane 2 will change to -500 after a short delay, which indicates that the cell is burnt out.
- **Ignition Rules**: a cell in the "fire spread plane" that has not ignited yet (i.e. has a value of 300 °K) will ignite if at least two of its neighbors have a plane 2 state of -200 or at least one neighbor with a plane 2 state of -300. The cell will ignite, and its temperature is updated to 301 °K.

3 ANALYSIS OF THE QUANTIZED MODEL FOR DIFFERENT QUANTUM SIZES

To analyze the effect of quantum size on the performance of the quantized version of the fire-spread Cell-DEVS model, we run both models using the CD++ simulator (López and Wainer 2004) to compare their results and analyze the size of the log files and the number of messages exchanged. The model was quantized using different quantum sizes: 0.7, 1.0 and 1.3 to generate different results and analyze the performance at each quantum size. As identified in (MacLeod, Chreyh, and Wainer 2006), we verified that the model takes long because the number of messages transferred between cells is very high. When we need to simulate a large area, this model will be inefficient. Faster simulation results were achieved when we run simulations with the quantized version. However, the quantum size influences the model's accuracy in addition to the simulation speed. Therefore, the target is to use quantization to reduce the execution time of the simulation without considerably reducing the accuracy.

The speed of the simulation and execution can be measured by analyzing the log files for each simulation process. The log file of the original model was 42MB in size and the number of messages exchanged was very large (503562).

Table 2: Quantum size vs. the exchanged number of messages

Quantum size	Total number of messages exchanged
0.7	233662
1.0	207412
1.3	123280

After quantizing the original model with quantum size 0.7, the number of messages exchanged was reduced to more than a half (i.e. to 233662 as shown in Table 2) and the log file size was reduced to 17 MB (more than a half too). As a result, simulation results were obtained faster.

To further test the improvement in the simulation speed, the model was executed with quantum sizes of 1 and 1.3, and the number of messages was 207412 and 123280, respectively (Table 2). Likewise, the size of the log files was reduced.

As seen above, the higher the quantum size the faster the simulation and the lower the number of messages exchanged and the log file size. However, there is a concern about accuracy as there is a chance of errors. A quantum size that achieves a trade-off between accuracy and execution speed must be selected. The impact of quantum size on the accuracy of the model is discussed below.

To analyze the difference, we can observe figures 4-7, which depict the results obtained from the original and the quantized models with different quantum sizes. One of the first things we can see is that the temperatures go much higher ("white hot") in the original model than in the quantized models. This indicates that the proposed quantized models are not as accurate as the original model because the temperature in the original model takes longer to fall down. This happens because the way the quantized model deal with border cells. Because the fire is in a confined space and most of the cells started out burning, there are fewer unburned cells to average out the temperature and cool the fire. This is analogous to the diffusion of heat in real scenarios.

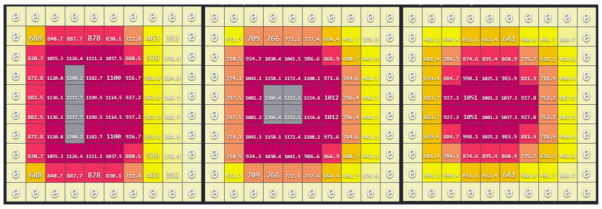

Figure 4: Fire Spread at 300, 700 and 1000.

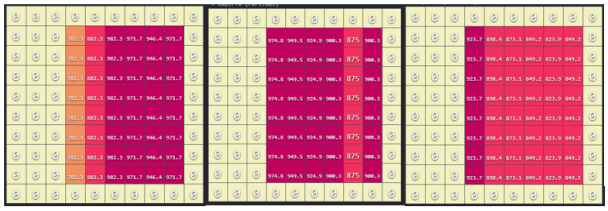

Figure 5: Quantized Fire Spread with quantum size 0.7 at 300, 700 and 1000.

As shown in figures 6 and 7, the fire spreading is quite different when quantized with higher quantum sizes. Because the quantized model follows a fixed curve, small inconsistencies in the initial state are magnified.

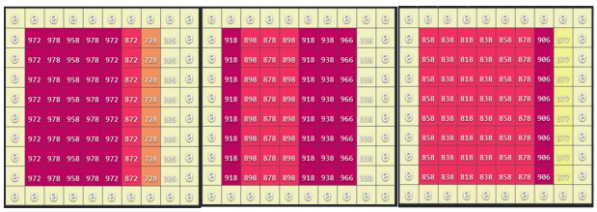

Figure 6: Fire Spread when quantized with 1 quantum size at 300, 700 and 1000.

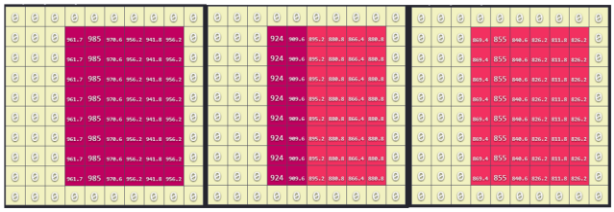

Figure 7: Fire Spread when quantized with 1.3 quantum size at 300, 700 and 1000.

In general, some of the quantized models have performed at a much better speed and produced much smaller logs, considering the issues mentioned above with the accuracy. Therefore, we need to find an acceptable quantum size that gives a trade-off between speed and accuracy.

4 MODIFYING THE QUANTIZED MODEL

We have defined a new version of the quantized model with an extended version of the CD++ simulator (López and Wainer 2004). This simulator has many advantages over the original version of the CD++ simulator that was used to implement the models in the previous sections. With the extended CD++ simulator, a cell can have multiple state variables and ports. As such, state variables can be used instead of using multiple layers (as in the previous models). This reduces the number of cells in the model, reduces the messages exchanged between the different layers, and hence improves the performance.

As with the previous models, the area is divided into a mesh of cells, with each cell having multiple variables (e.g., temperature, ignition time, status, etc.) and the quantum size is defined using an external file. The neighborhood in this model is the Von Neumann neighborhood containing cells in the same layer only (as this model has one layer). The temperature is stored in a variable temp and is sent to the default port (out), while the second layer was substituted by a port (info). As mentioned above, by doing so, the numbers of cells and messages to send during simulation is reduced. The third plane is deemed unnecessary and hence eliminated (the information on this plane was never used in the simulation). The minimum quantum size was set to 0.43 K as it provides results in a reasonable execution time (8.842 seconds) and the results generated are close to the original model as discussed later in section 4.2.

In the following, we list the rules used in this model (the same as in the quantized version in section 2.2) along with a code snippet from the implementation of the rules. The code below is written using a language built in the CD++ simulation engine to define the rules for the cells.

- A cell in the burning up phase

```
rule : {~temp := (0,0)~temp + #macro(q); } {round (((11.56 * exp( 0.0005187 * (
(0,0)~temp + #macro(q) ) ) - 784.7 * exp(-0.01423 * ( (0,0)~temp + #macro(q) ) ) )  -
(11.56 * exp( 0.0005187 * (0,0)~temp ) - 784.7 * exp(-0.01423 * (0,0)~temp ) ) ) *
100)} {  ( (0,0)~status = -100) or ( (0,0)~status = -200) or ( (0,0)~status = -300 ) }
```

- A cell in the burning down phase

```
rule : {~temp := (0,0)~temp - #macro(q); } {round ( #macro(q) * 0.052 * 100 ) } { (
(0,0)~status = -400) }
```

- A cell whose state in plane 2 is 0 and has a temperature between 301 and 474:

```
rule : {~status := -100; } 1 {((0,0)~temp >= 301) and ((0,0)~temp < 474) and
((0,0)~status = 0) }
```

- A cell whose state in plane 2 is 0 or -100, and has a temperature greater than or equal to 474

```
rule : {~status := -200; } 1 {((0,0)~temp >= 474) and ( ( (0,0)~status = -100) or (
(0,0)~status = 0 ) ) }
```

- A cell whose state in plane 2 is 0 or -200, and has a temperature greater than or equal 650

```
rule : {~status := -300; } 1 {((0,0)~temp >= 650) and ( ((0,0)~status = -200) or
((0,0)~status = 0) ) }
```

- A cell whose state in plane 2 is 0 or -300, and has a temperature greater than or equal 992

```
rule : {~status := -400; } 1 {((0,0)~temp >= 992) and ( (0,0)~status = -300)}
```

- A cell whose state in plane 2 is -400, and its temperature is less than or equal 333

```
rule : {~temp := 273; ~status := -500; } 1 {((0,0)~temp <= 333) AND ( (0,0)~status = -
400 )}
```

- Ignition Rules

```
rule : {~temp := 301; ~status := -100; } 1 { ((0,0)~status = 0) AND (0,0)~temp = 300
AND stateCount ( -200, ~status ) >= 2}
rule : {~temp := 301; ~status := -100; } 1 { ((0,0)~status = 0) AND (0,0)~temp = 300
AND stateCount ( -300, ~status ) >= 1}
```

We will show three sets of simulations performed with the improved model. The first simulation (Group of Fire Lines) used the same initial conditions as the experiment carried out in Section 3. The results of the simulation are compared with the basic and the quantized models. The second set of simulations implements the model in a case where there is only one fire line in the middle and fire must spread to both sides (Middle Fire Line). Finally, the third set of simulations considers a scenario where there are fire lines in the extreme columns of the cell space (Extreme Fire Lines).

The initial conditions for the simulation for the three models are shown in figure 8. Results at 100, 300, and 500 are shown in figures 9 to 11. As can be seen in the figures, the proposed model has a small additional error in early stages of the simulation since it shows generally lower temperatures than the other two models. However, after 300 it can be noticed that the proposed model produces closer results to the original model than the quantized model, especially in the near-border columns. The proposed model increases the temperatures relatively slower in the beginning, but the temperatures tend to be more accurate than the quantized model as the simulation progresses. Furthermore, the quantized model has high temperatures for all the cells, and the proposed models maintains the tendency of the original model of higher temperatures near the center, and lower in the outer cells. This general behavior is maintained throughout the simulation, especially for the outer columns. Finally, it is important to mention that the log file of the proposed model had only 151,885 messages, compared to the 966,253 messages produced by the quantized model.

Figure 8: Initial conditions for the "group of fire lines" simulation.

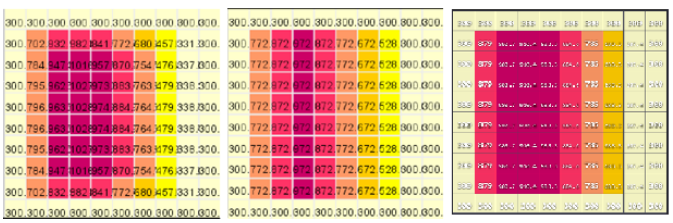

Figure 9: From left to right, results at 100 of the basic, quantized, and proposed models.

Figure 10: From left to right, results at 300 of the basic, quantized, and proposed models.

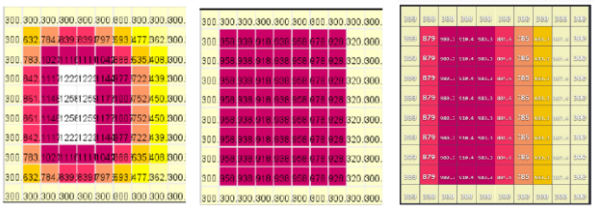

Figure 11: From left to right, results at 500 of the basic, quantized, and proposed models.

Figure 12 shows the results for the "middle fire line" simulations at times 0 (initial condition), 3000 and 5000. The model produces the desired behavior, since the temperatures of the central columns start increasing, and the temperature of the columns next to the central line increases progressively. At time 5000, the central column has been burnt completely and all the other columns are either burning up or down. The border cells are set to maintain their temperature at all times and they do so, as seen in the figure.

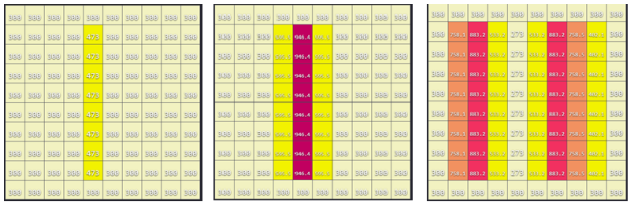

Figure 12: "Middle fire line" simulations of the proposed model at times 0, 3000 and 5000.

Figure 13 shows the results for the "near-boarder fire lines" simulations at times 0 (initial condition), and t = 2000, 4000, and 5000. The model behaves as it was intended since the temperature of the near-boarder columns increases and causes the central columns' temperature to increase. At time 5000, the near-boarder columns are completely burnt, and every other column (except the borders) is increasing or decreasing its temperatures.

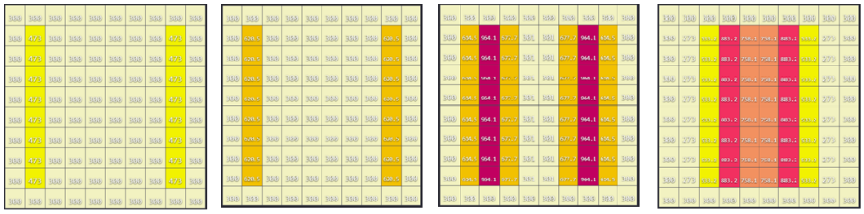

Figure 13: "Near-border fire lines" simulations of the proposed model at times t = 2000, 4000, and 5000.

5 CONCLUSION

Modeling and Simulation provides an efficient and convenient tool to study and analyze the phenomena of fire spread. We introduced two fire spread Cell-DEVS models. In the Quantized fire model, the basic fire spread model was modified to achieve a faster simulation without a considerable loss of accuracy. We use these models to study the impact of quantum size used in the quantized fire spread model on the

accuracy and performance of the model. We measured the performance of the model based on the number of messages exchanged and the size of the simulation log file.

Moreover, we developed an improved model definition and implementation of the quantized model using a newer and extended version of the CD++ simulator. The proposed model eliminates the need for using additional planes in the model, which reduces the number of exchanged messages between cells and improves execution time. Results show that the proposed model does not just improve execution time over the original quantized model, but also produces more accurate results.

6 REFERENCES

Albini, F.A. 1985. A Model for Fire Spread in Wildland Fuels By-Radiation†. *Combustion Science and Technology* 42, no. 5–6 (March 21): 229–258. http://www.tandfonline.com/doi/abs/10.1080/00102208508960381.

Balbi, J.H., P.A. Santoni, and J.L. Dupuy. 1999. Dynamic Modelling of Fire Spread across a Fuel Bed. *International Journal of Wildland Fire* 9, no. 4: 275–284. http://www.publish.csiro.au/?paper=WF00005.

Filippi, J.-B., and P. Bisgambiglia. 2004. JDEVS: An Implementation of a DEVS Based Formal Framework for Environmental Modelling. *Environmental Modelling & Software* 19, no. 3 (March 1): 261–274.

Fons, W.L. 1946. Analysis of Fire Spread in Light Fuels. *Journal of Agricultural Research* 72 (January): 93–121.

Hwang, C.C., and Y. Xie. 1984. Flame Propagation Along Matchstick ArraysOn, Inclined Base Boards. *Combustion Science and Technology* 42, no. 1–2 (December 21): 1–12. http://www.tandfonline.com/doi/abs/10.1080/00102208408960366.

López, A., and G. Wainer. 2004. Improved Cell-DEVS Model Definition in CD++. In *Cellular Automata ACRI 2004. Lecture Notes in Computer Science, Vol 3305*, 803–812. Berlin, Heidelberg: Springer.

MacLeod, M., R. Chreyh, and G. Wainer. 2006. Improved Cell-DEVS Models for Fire Spreading Analysis. In *Cellular Automata ACRI 2006. Lecture Notes in Computer Science, Vol 4173*, 472–481. Berlin, Heidelberg: Springer.

McArthur, A.G. 1966. Weather and Grassland Fire Behaviour. *Forestry and Timber Bureau*no. 100.

McGrattan, K., R. McDermott, J. Floyd, S. Hostikka, G. Forney, and H. Baum. 2012. Computational Fluid Dynamics Modelling of Fire. *International Journal of Computational Fluid Dynamics* 26, no. 6–8 (July 26): 349–361. https://www.tandfonline.com/doi/full/10.1080/10618562.2012.659663.

Muzy, A., E. Innocenti, A. Aiello, J.-F. Santucci, and G. Wainer. 2005. Specification of Discrete Event Models for Fire Spreading. *SIMULATION* 81, no. 2 (February 18): 103–117.

Muzy, A., E. Innocenti, A. Aïello, J.F. Santucci, P.-A. Santoni, and D.R.C. Hill. 2004. Dynamic Structure Cellular Automata in a Fire Spreading Application. *Undefined*. https://www.semanticscholar.org/paper/Dynamic-Structure-Cellular-Automata-in-a-Fire-Muzy-Innocenti/5522389ba6063aa908457159d7a48f12e6cd402e.

Novozhilov, V. 2001. Computational Fluid Dynamics Modeling of Compartment Fires. *Progress in Energy and Combustion Science* 27, no. 6 (January 1): 611–666. https://www.sciencedirect.com/science/article/pii/S0360128501000053.

NRC. 2019. Forest Fires. Accessed January 4. https://www.nrcan.gc.ca/forests/fire-insects-disturbances/fire/13143.

Pastor, E., L. Zárate, E. Planas, and J. Arnaldos. 2003. Mathematical Models and Calculation Systems for the Study of Wildland Fire Behaviour. *Progress in Energy and Combustion Science* 29, no. 2 (January 1): 139–153. https://www.sciencedirect.com/science/article/pii/S0360128503000170.

Pingitore, G. 2017. *Community and Protective Services Committee*. Ottawa, Canada. https://documents.ottawa.ca/sites/default/files/annual report 2017 en.pdf.

Rothermel, R.C. 1972. A Mathematical Model for Predicting Fire Spread in Wildland Fuels. *Res. Pap. INT-115. Ogden, UT: U.S. Department of Agriculture, Intermountain Forest and Range Experiment Station. 40 P.* 115.

Sarjoughian, H.S., and B.P. Zeigler. 2014. DEVSJAVA : Basis for a DEVS-Based Collaborative M&S

Environment. *Simulation Series* 30: 29–36.

Vicino, D., D. Niyonkuru, G. Wainer, and O. Dalle. 2015. Sequential PDEVS Architecture. In *Symposium on Theory of Modeling & Simulation: DEVS Integrative M&S Symposium*, 1–12. Alexandria, USA.

Wainer, G.A. 2009. *Discrete-Event Modeling and Simulation : A Practitioner's Approach*. Boca Raton: CRC Press.

Wainer, G.A., and N. Giambiasi. 2002. N-Dimensional Cell-DEVS Models. *Discrete Event Dynamic Systems* 12, no. 2: 135–157.

Wainer, G.A., and B.P. Zeigler. 2000. Experimental Results of Timed Cell-DEVS Quantization. In *IN PROCEEDINGS OF AIS'2000*, 1–7.

Zeigler, B.P., H. Praehofer, and T.G. Kim. 2000. *Theory of Modeling and Simulation: Integrating Discrete Event and Continuous Complex Dynamic Systems*. San Diego: Academic Press.

AUTHOR BIOGRAPHIES

ALA'A AL-HABASHNA has obtained his Master of Engineering degree in Electrical and Computer Engineering from Memorial University of Newfoundland, St. John's, Canada, and his PhD in Electrical and Computer Engineering from Carleton University, Ottawa, Canada. Currently, he is a Postdoctoral Fellow at the Department of Systems and Computer Engineering at Carleton University. His email address is alaaalhabashna@sce.carleton.ca.

CRISTINA RUIZ-MARTIN has obtained a Ph.D. in Industrial Engineering (University of Valladolid, UVa) and Systems and Computer Engineering (Carleton University). She is a Postdoctoral Fellow at the Department of Systems and Computer Engineering at Carleton University. Her email address is cristinaruizmartin@sce.carleton.ca.

GABRIEL WAINER is a Professor at the Department of Systems and Computer Engineering at Carleton University. He is a Fellow of SCS. His email address is gwainer@sce.carleton.ca.