Modeling Forest Fire Using Cell-DEVS

Jingqi Zhang
Carleton University
1125 Colonel By Drive
Ottawa, ON. K1S-5B6 Canada

ABSTRACT

Modeling and simulation of forest fires is critical for disaster prediction and resource scheduling. The traditional Corsica model is based on Cell-DEVS and uses a synchronous update mechanism, but the simulation efficiency is low due to frequent message passing. In this study, we propose to combine the Quantized DEVS (Q-DEVS) method to significantly reduce the simulation time while maintaining reasonable accuracy by dynamically adjusting the quantum threshold and optimizing the neighbor communication.

Keywords:Discrete Event System Specification (DEVS), Cell-DEVS, Particles transport, Modeling, Simulation, CD++.

1. INTRODUCTION

Forest fires are one of the global natural disasters, which not only threaten the stability of ecosystems, but also pose a huge threat to the safety of human life and property. With climate change and the intensification of human activities, the frequency and intensity of forest fires are on the rise. Therefore, the establishment of an efficient and accurate forest fire modeling and simulation system has become a core requirement in disaster early warning and emergency management.

Modeling and simulation technology can provide a scientific basis for decision-makers to optimize the scheduling and deployment of fire extinguishing resources and improve the efficiency and accuracy of emergency response by predicting the development trend, spread path and scope of fire before the actual fire occurs. At present, the Corsica model based on the Cell-DEVS framework is widely used in the field of forest fire modeling. By defining local rules, the model simulates key physical processes such as heat transfer and combustion propagation in detail, which provides an important reference for understanding fire behavior.

However, the Corsica model adopts a synchronous update mechanism, which requires a large number of information exchanges between units at each time step, resulting in huge system communication overhead and low simulation efficiency. This feature is acceptable in small-scale simulations, but in large-scale, complex terrain or high-resolution scenarios, real-time and scalability problems become more and more prominent, which seriously limits its application potential in practical disaster management.

In order to solve the above problems, improving the efficiency and scalability of forest fire modeling and simulation has become an important direction of current research. Optimizing the model structure, improving the update mechanism, and reducing redundant communication can not only improve the performance of large-scale simulation, but also enhance the adaptability and practical value of the model in the actual emergency response system.

Cellular Automata (CA) is a well-known formalism. It was originally introduced by von Neumann to study self-reproducing systems. All the models [3,4,5] I referenced in this project apply CA as a basic methods. In CA, states in the lattice are updated according to a local rule in a simultaneous, synchronous way, and cell states change in discrete time steps [1]. Cell-

DEVS [1,] is another formalism which is used to capture the behaviour of systems that can be represented as cell spaces. Cell-DEVS has explicit time delay, which is different form CA. CD++ [1] is a toolkit designed to perform the simulations which specified by DEVS or Cell-DEVS formalisms. This project is about using Cell-DEVS formalism to model forest fire systems, and quantitative methods are used to reduce the amount of simulation information transferred to improve the simulation performance of the system

2. BACKGROUND

2.1 Discrete Event System Specification (DEVS)

DEVS is formalism for modeling and analysis of discrete event systems, which was invented by Dr. Bernard P. Zeigler in 1976. DEVS provides a formal model and simulation framework, which separates modeling from simulation and provides mechanisms of defined coupling of components. DEVS Framework for M&S supports dynamic system representation and hierarchical, modular model development, which is able to model both continuous and discrete time systems [Zeigler, 1976/84/90/00]. Since Dr. Zeigler proposed a hierarchical simulation algorithm for DEVS model simulation in 1984, DEVS formalism had been widely used, and many extended formalisms from DEVS have been developed. Cell-DEVS formalism is one of them, which combines DEVS and CA. DEVS Formalism consists of two models, DEVS Atomic models and DEVS coupled models. Therefore, Basic DEVS atomic models that can be coupled into some DEVS coupled models hierarchically to build complex simulations.

2.2 Cell-DEVS: Cell-DEVS is an extension of DEVS by adopting the concept of cell space. The concept of cell Spaces is shown by Figure 1. Cell space can be specified by the following formalism. CCA = < S, n, C, η, N, T, t > C cell's state variables; S finite alphabet to represent each cell's state; n dimensional space; N neighboring cells; T global transition function; t local computing function.

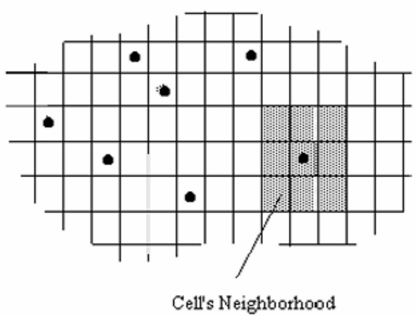

Figure 1 cell space and neighbourhood

Each cell has N inputs form N neighbors. When an external event occurs, the local computing function is activated and executed,and the inputs are consumed by the cell.

Commonly used rectangular neighborhoods are von Neumann Neighbors and Moore Neighbors. Certainly, regular or irregular neighborhood can be specified as need.

Cell-DEVS has explicit timing delays which can be either transport delays or inertial delays. Transport delay is shown in the Figure 2.

In Cell-DEVS, an individual cell is considered as an atomic model, and its corresponding cell space is defined as a coupled model. The basic model of Cell-DEVS is atomic cell model, which automatically couple between the cells.

A coupled Cell-DEVS model might be decomposed into several cell atomic models and some other DEVS atomic models or coupled models. Figure 5 shows a coupled Cell-DEVS model informally.

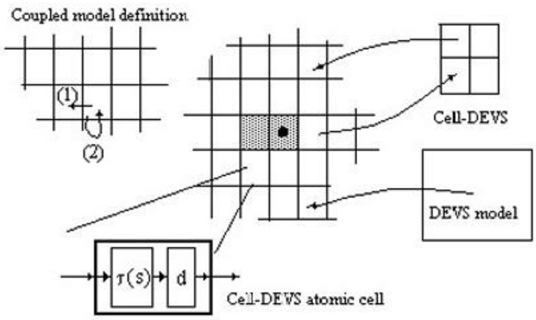

Figure 2

2.3 CD++

CD++ tool [1,7,8] has been developed following the specifications of DEVS and Cell-DEVS. In the toolkit, it includes many extensions to support integrations with other simulation environments, and provide services to build models based on DEVS or Cell-DEVS formalism, to execute simulations, and to visualize the results obtained in the simulation. The simulation engine of CD++ tool is built as a class hierarchy. There two versions of CD++ were developed for simulating Cell DEVS models. One is commonly used to simulate general Cell DEVS models. However, a huge number of neighbors are required, if a Cell-DEVS model has multiple state variables for each cell. The execution time increase significantly. Another version of CD++ was developed for this purpose in [7]. Manipulations of multiple state variables are done within each cell in each time step. It also supports multiple inter-cell ports to communicate with the neighbours, and improves the definition of complex models. Another significant advantage of second version is the execution time of simulations reduced dramatically.

3. MODEL DEFINITION

Cell-DEVS models

The improved model uses an extended atomic model definition to integrate Q-DEVS quantification with humidity extinguishing strategies:

The forest fire model is based on the Cell-DEVS formalization framework, which discretizes the forest area into a rectangular grid, with each cell representing an independent plot. The model simulates the ignition, spread, and interaction with the environment (such as wind and humidity) through local rules, aiming to predict the dynamic behavior of the fire and optimize the emergency response strategy.

Model definition:

X = Ø Y = Ø S = temp (Kelvin) - T $\in R_0^+$ fire_status = {0,1,2,3,4} t_ig – t $\in R_0^+$ sigma – t $\in R_0^+$ quantum=Q$\in R_0$ humidity=H$\in R_0$

```
N = {(-1,-1), (-1, 0), (-1, 1),
         (0, -1), (0, 0), (0, 1),
         (1,-1), (1, 0), (1, 1)}      (Moore's Neighourhood)
d = 1 time unit
τ = Rule 1: (Border cells stay constant and passivate)
                  if s.fire_status = 0 then:
                       s.temp = 300
                       s.sigma = ∞
      Rule 2: (inactive cells remain inactive until temp changes)
                  if s.fire_status = 1 then:
                       Update s.temp using eq. (1)
                        If new s.temp = 300 then:
                                   s.sigma = ∞
                        Else:
                                   s.fire_status = 2
                                   s.sigma = 1.0
      Rule 3: (active cells remain active until the reach 300K (passivate) or 573K (ignite))
                  if s.fire_status = 2 then:
                        If s.temp < 573 then:
                               Update s.temp using eq. (1)
                                If new s.temp = 300 then:
                                          s.fire_status = 1
                                          s.sigma = ∞
                                Else:
                                          s.sigma = 1.0
                        Else if s.temp >= 573 then:
                                  Update s.temp using eq. (1)
                                   s.fire_status = 3
                                   s.sigma = 1.0
                                   s.t_ig = 1.0            (set t_ig for time t+1)
      Rule 4: (Cells burn using enthalpy term until they drop to 333K)
                  if s.fire_status = 3 then:
                        If s.temp > 333:
                               Update s.temp using eq. (1)
                                  s.sigma = 1.0
                                  s.t_ig += 1.0
                        Else if s.temp <= 333:
                                  Update s.temp using eq. (1)        (without enthalpy term)
```

```
                                    s.fire_status = 4
                                    s.sigma = 1.0
        Rule 5: (Cells cool until they reach 300K, then passivate)
                        if s.fire_status = 4 then:
                              Update s.temp using eq. (1)
                              If new s.temp = 300:
                                    s.sigma = ∞
                              Else:
                                    Update s.temp using eq. (1)    (without
enthalpy term)
                                    s.sigma = 1.0

δint, δext, λ and ta are defined using Cell-DEVS specifications.
```

In order to improve the efficiency and accuracy of forest fire modeling and simulation, this paper proposes three key improvement measures based on quantum triggering, wind direction fusion and humidity influence to optimize the performance of the simulation system and improve its adaptability to the actual environment, aiming at the problems of redundant calculation, communication congestion and lack of real-time performance of traditional synchronous update mechanism in large-scale applications.

the Quantum Trigger Rule[2] sets a quantum threshold based on temperature change to address the drawbacks of traditional models that forcibly update the cell state and frequently exchange information at each time step. When the temperature change in the cell does not exceed the set threshold, the state update and message transmission are not triggered, and only significant changes trigger communication and calculation. This mechanism effectively filters small changes with unclear physical significance, significantly reduces the system communication volume and computing overhead, thereby greatly improving the overall simulation efficiency while ensuring the simulation accuracy, which is particularly suitable for high-resolution and wide-area forest fire simulation tasks.

the Wind-Integrated Propagation Rule introduces a dynamic neighbor weight adjustment mechanism to systematically integrate wind speed and wind direction factors into the modeling of the fire spread process. Specifically, according to the current wind direction, the weight of the neighboring unit in the downwind direction is moderately enhanced, thereby increasing its heat contribution to the central unit; while the influence of the neighboring unit in the upwind direction is correspondingly weakened. The greater the wind speed, the more obvious the weight adjustment. This design accelerates the spread of fire in the wind-dominated direction while suppressing the spread in the upwind direction, truly reflecting the important modulation effect of meteorological conditions on the dynamic behavior of fire, and improving the accuracy and credibility of the model's prediction of the path and speed of fire spread.

the Humidity Modulation Rule further enriches the physical properties of the unit cell, and dynamically affects the combustion state and activation threshold of the unit cell by introducing humidity parameters. The higher the humidity, the more heat the unit cell needs to accumulate before it can be ignited, and the burning rate is relatively reduced; while the unit cell with lower humidity is easier to ignite, and the burning process is more intense. This rule not only enhances the applicability of the model under different landforms, vegetation types, and seasonal changes, but also makes the fire simulation more detailed and realistic, and can simulate complex phenomena such as the formation of a "fire isolation zone" in a local humid area.

By adding the judgment condition of the quantum threshold, the cells below the quantum threshold are modified to a dormant state, and only the cells that exceed the quantum threshold will propagate signals to the surrounding neighbors, and the specific implementation code is as follows

```
if (state.fire_status != 0) {
    double delta = new_temp - old_temp;
    if (std::abs(delta) >= state.quantum) {
        state.temp = new_temp;
        state.sigma = 1.0;
    } else {
        state.sigma = std::numeric_limits<double>::infinity();
    }
}
```

At the same time, the humidity factor is added on the basis of the original model, and the cells will not be ignited within a certain humidity range, and when the humidity exceeds a certain range, even if the cells are burned, they will also move to the next state. Humidity is used to simulate the impact of phenomena such as non-flammable areas or man-made firewalls on the spread of fires in the event of a fire.

4. Simulation result and discussion

4.1 quantization method

In order to quantitatively evaluate the effect of the quantization method in reducing the amount of simulation calculations, this paper selected three groups of different quantum thresholds for comparative experiments under the same scenario conditions. The experiments all adopted the configuration of wind direction 90° and wind speed 15 mph set in the original model to ensure the comparability and consistency of the comparison results

when the quantum threshold is set to 0.3, the size of the output file generated by the simulation is reduced by about 10 MB compared with the original model, accounting for about 1.42% of the original file size. This result shows that even if a smaller quantum threshold is set, a certain degree of communication volume and storage space optimization can be achieved while ensuring high simulation accuracy.

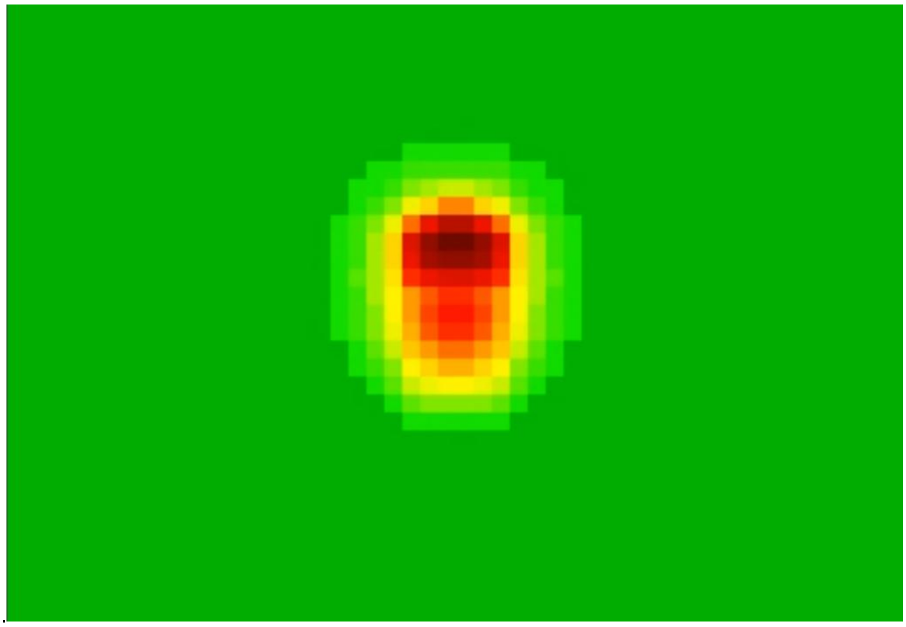

Figure 3 simulation result

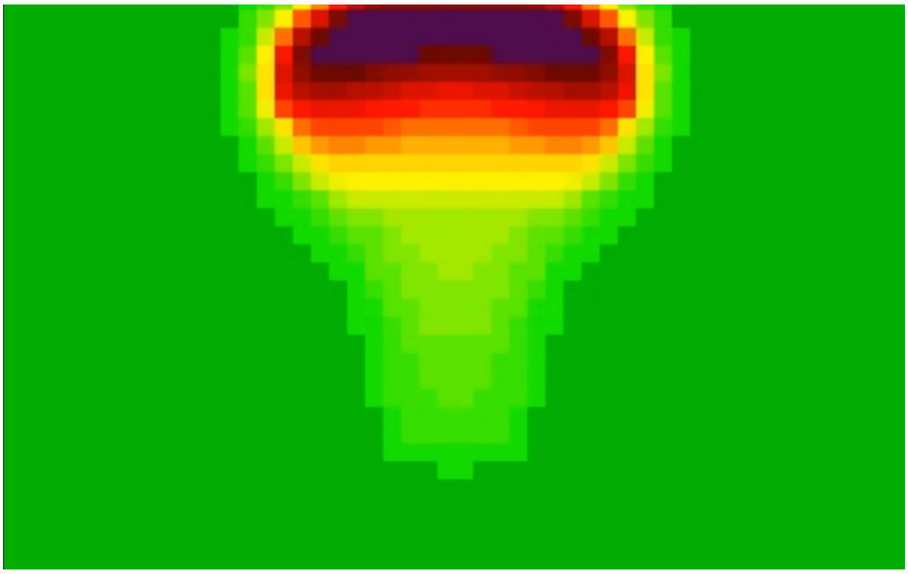

Figure 4 simulation result

When the quantum threshold is increased to 2, the size of the generated file is reduced by 40 MB compared with the original model, which is equivalent to a reduction of 5.7%. At this time, due to the increase in the threshold, more cells enter a dormant state when the temperature change is insufficient, thereby further reducing the number of message transmission and status updates. However, the increase in the number of dormant cells also leads to the loss of details in the simulation process, the dynamic accuracy of the fire spread is reduced, and the details of the local fire expansion cannot be accurately captured.

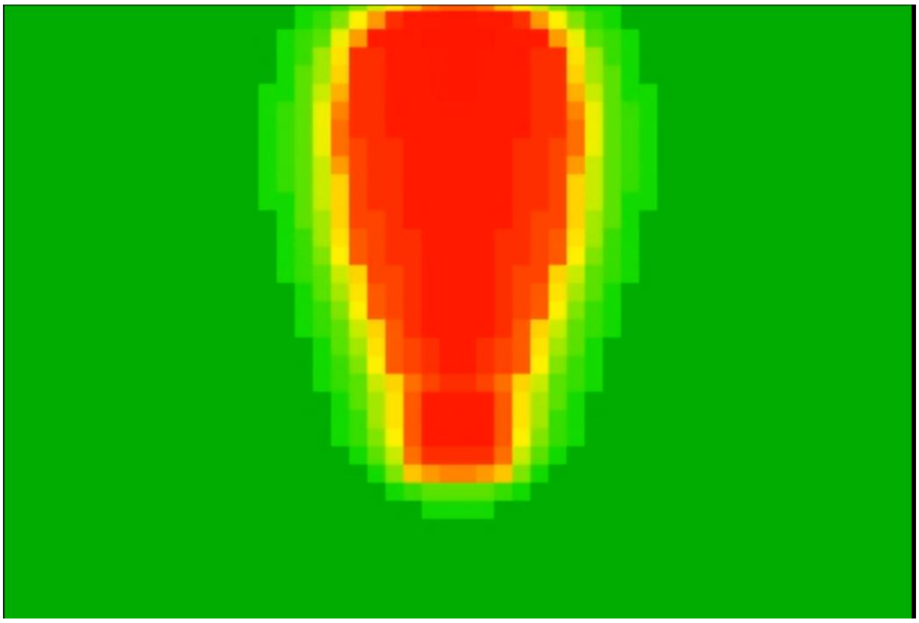

Figure 5 quantum=2

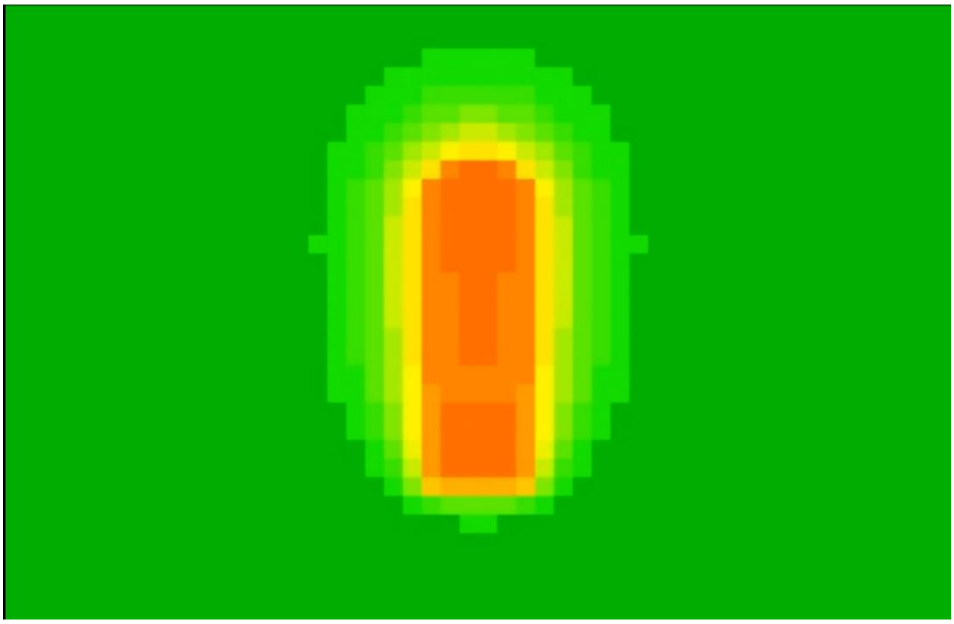

Figure 6 quantum=2.5

When the quantum threshold increases to 2.5, the simulation accuracy continues to decline, the spatial continuity and temporal coherence of the fire spread are more affected, and the local burned area becomes incoherent or lagging. Finally, in the extreme case where the quantum threshold is set to 30, almost all cells are in a dormant state for a long time because the temperature change fails to meet the judgment conditions, resulting in the failure of the fire to spread normally, and the simulation results are seriously distorted, basically losing their actual reference value

In summary, the quantization method can effectively reduce the computational load and storage requirements of the simulation system within an appropriate threshold range, but the selection of the quantum threshold needs to be balanced between simulation accuracy and performance optimization. Although an excessively high quantum threshold brings more significant performance improvements, it inevitably leads to a significant decrease in model accuracy, limiting the reliability and practical application value of the simulation results.

.4.2Humidity factor

In the simulation configuration, setting up firewalls (Firebreaks) or non-combustible zones (Non-combustible Zones) can effectively simulate the situation of blocking the spread of fire in real terrain. For example, rivers, bare rock areas, artificial isolation zones, etc. can be regarded as natural or man-made non-combustible obstacles. Introducing such elements into the model can not only change the path and speed of fire spread, but also simulate the complex behaviors of fire when encountering obstacles, such as detours, slowing spread, or partial extinguishing. By constructing a more realistic scene setting, the simulation results can more accurately reflect the impact of different prevention and control measures on the development of fire, and provide strong data support and decision-making reference for the formulation of effective fire extinguishing strategies, fire isolation zone layout planning and emergency response plans. This method greatly improves the application value of simulation, making the model not only have predictive capabilities, but also serve the actual disaster management and prevention and control strategy design.

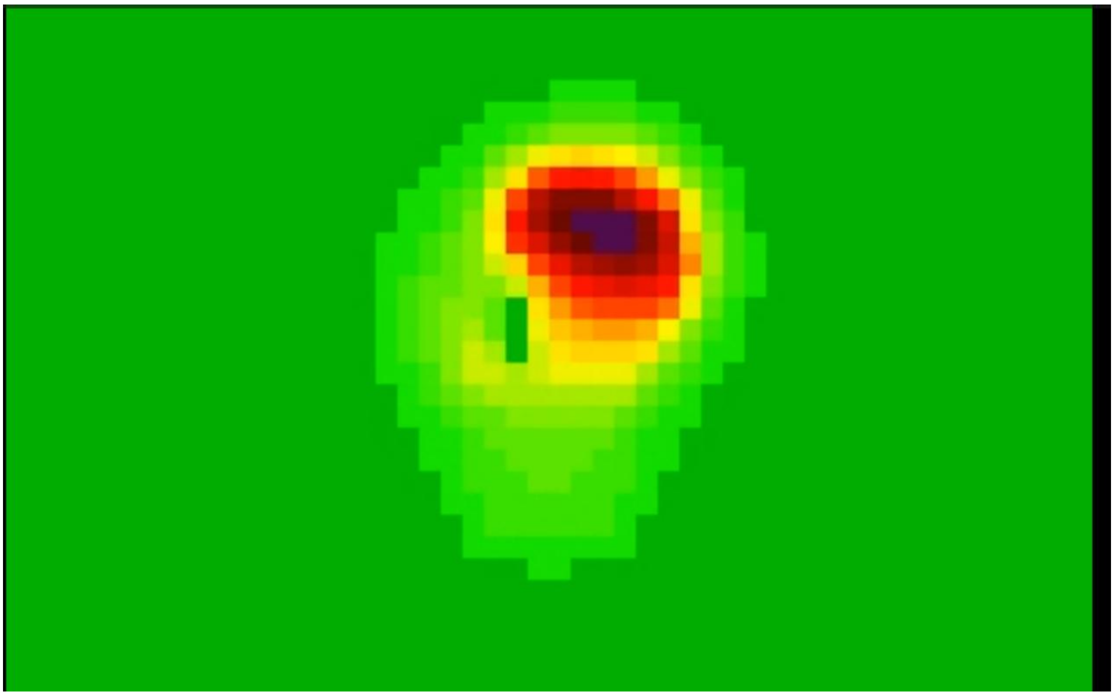

Figure 7 simulation with wet zone

In the figure, the middle area represents a non-combustible region, meaning it cannot catch fire or contribute to the spread of flames. The presence of this non-combustible area significantly alters the overall fire dynamics compared to a fully combustible environment. Instead of a uniform outward spread, the fire is forced to propagate around the obstacle, resulting in asymmetrical flame fronts, delayed expansion in certain directions, and complex flow patterns influenced by the obstruction. This highlights how variations in terrain composition, such as bodies of water, rocky surfaces, or firebreaks, can critically impact fire behavior and must be accurately represented in fire simulation models to achieve realistic predictions.

5.Future work

5.1. Multi-factor coupling modeling

In the future, more key environmental variables, such as precipitation, vegetation type, terrain slope, soil moisture, etc., can be introduced to establish a multi-factor coupling fire propagation model. By comprehensively considering the complex interaction of climatic conditions and terrain characteristics, the model will be able to more carefully depict the dynamic changes of fire spread under different scenarios, and improve the simulation accuracy and adaptability of fire behavior in extremely complex landforms and changeable meteorological environments.

5.2. Actual case verification

In order to further verify the actual application effect of the model, the improved model should be applied to real historical fire cases, such as typical events such as the Australian bushfire and the California wildfire. By comparing the simulation results with the actual fire development process, adjusting and calibrating the model parameters, and evaluating its prediction accuracy and robustness. This not only helps to test the applicability of the model under different geographical and climatic conditions, but also provides a reference decision support tool for actual disaster management and enhances the engineering application value of the model.

5.3. Interactive visualization tool development

In order to improve the user experience and application efficiency of the model, an interactive visualization platform can be developed in the future to integrate simulation operation, parameter adjustment and result rendering functions. The platform will support users to modify input parameters (such as wind speed, wind direction, humidity changes, etc.) in real time and dynamically observe the fire spread process, providing a more intuitive and flexible analysis method. Through the graphical interface, decision makers and emergency responders can more easily understand the fire spread trend, quickly formulate targeted prevention and control and emergency measures, and greatly improve decision-making efficiency and response speed.

6.Conclution

In view of the high communication overhead and low simulation efficiency of the traditional Corsica forest fire model under the synchronous update mechanism, this paper proposes an improved method based on the Cell-DEVS framework and combined with the quantized discrete event system (Q-DEVS) to improve the performance and accuracy of large-scale fire simulation. In view of the redundant calculation and information transmission bottlenecks commonly found in traditional models, this paper designs and introduces three core improvement mechanisms: the Quantum Trigger Rule, the Wind-Integrated Propagation Rule, and the Humidity Modulation Rule.

Specifically, the Quantum Trigger Rule sets a dynamic threshold for temperature changes, triggering state updates and message transmission only when the threshold is exceeded, effectively filtering out small and meaningless changes, thereby significantly reducing the communication volume and computational burden during the simulation process. The Wind-Integrated Propagation Rule adjusts the heat weight of neighboring cells based on real-time wind direction and wind speed, making the direction and speed of fire expansion more consistent with changes in meteorological conditions, and improving the model's ability to respond to dynamic changes in the external environment. The humidity regulation rule accurately reflects the heterogeneous characteristics of fire spread under different humidity environments by introducing cell humidity properties, regulating combustion conditions and activation thresholds, and significantly enhancing the physical reality and local detail restoration capabilities of the simulation, especially when simulating non-combustible areas (such as rivers and firewalls).

The experimental results show that within a reasonable quantum threshold range (such as 0.3 to 2), the model calculation load is significantly optimized, and the output file size is reduced by about 1.42% to 5.7%. At the same time, the fire spread path maintains a high consistency with the original model, and the simulation accuracy is basically guaranteed. However, when the quantum threshold is set too high (such as 30), a large number of cells enter dormancy, resulting in the inability of the fire to spread normally and severe simulation distortion, which verifies the importance of quantization parameter selection for the balance between model performance and accuracy.

Overall, the improved method proposed in this paper effectively improves the efficiency and environmental adaptability of forest fire modeling and simulation, especially in large-scale complex scenarios, showing good scalability and application potential. In future work, we can further explore multi-scale coupled simulation, heterogeneous terrain modeling and fire evolution prediction under dynamic meteorological conditions, and provide more accurate and efficient technical means for forest fire prevention, emergency response and disaster decision support.

7. REFERENCES

[1] Wainer, Gabriel, A. and Mosterman, P. J. Discrete-Event Modeling and Simulation: Theory and Applications. Boca Raton: CRC Press.

[2] M. MacLeod, R. Chreyhm, “Project Report: Improving the Corsican Fire Spreading Cell-DEVS Model.” Submitted for SYSC 5104, Carleton University, May 2005.

[3] Masselot, A and Chopard, B. 1996. Cellular automata modeling of snow transport by wind’, Applied Parallel Computing Computations in Physics, Chemistry and Engineering Science, Lecture Notes in Computer Science, Volume 1041/1996, 429-435. DOI= http://doi.acm.org/0.1007/3-540-60902-4_45

[4] Tsunematsu, K, Falcone, J.L, Bonadonna, C, and Chopard, B. 2008. Applying a Cellular Automata Method for the Study of Transport and Deposition of Volcanic Particles. Cellular Automata. Lecture Notes in Computer Science, 2008, Volume 5191, 393-400. DOI= http://doi.acm.org/10.1007/978-3-540-79992-4_51

[5]Masselot, A and Chopard, B. 1998. A lattice Boltzmann model for particle transport and deposition. Europhys. Lett. 42 259 DOI= 10.1209/epl/i1998-00239-3