SIMULATION
Transactions of The Society for Modeling and Simulation International

DEVS-FIRE: Towards an Integrated Simulation Environment for Surface Wildfire Spread and Containment

Journal:	*Simulation: Transactions of the Society for Modeling and Simulation International*
Manuscript ID:	A-07-0066
Manuscript Type:	1 Applications
Date Submitted by the Author:	05-Sep-2007
Complete List of Authors:	Ntaimo, Lewis; Texas A&M University, Department of Industrial Engineering Hu, Xiaolin; Georgia State University, Computer Science Department Sun, Yi; Georgia State University, Computer Science Department; Georgia State University, Computer Science
Key Words and Phrases:	Applications in Science and Engineering -> Ecological and Environmental Systems , Theory and Methodology -> DEVS Methodology , Tools and Technology -> Agent-based Systems , Tools and Technology -> Modeling and Simulation Environments , Tools and Technology -> Simulation System Architecture

DEVS-FIRE: Towards an Integrated Simulation Environment for Surface Wildfire Spread and Containment

Lewis Ntaimo
Department of Industrial and Systems Engineering
Texas A&M University, College Station, TX, 77843
ntaimo@tamu.edu

Xiaolin Hu and Yi Sun
Department of Computer Science
Georgia State University, Atlanta, GA, 30303
xhu@cs.gsu.edu, ysun9@student.gsu.edu

Simulating wildfire spread and containment remains a challenging problem due to the complexity of fire behavior. In this paper, the authors present an integrated simulation environment for surface wildfire spread and containment called DEVS-FIRE. DEVS-FIRE is developed based on the discrete event system specification (DEVS) and uses a cellular space model for simulating wildfire spread and agent models for simulating wildfire containment. The cellular space model incorporates real spatial fuels data, terrain data, and temporal weather data into the prediction of wildfire behavior across both time and space. DEVS-FIRE is designed to be integrated with stochastic optimization models that use the scenario results from the simulation to determine an optimal mix of firefighting resources to dispatch to a wildfire. Preliminary computational experiments with fuel, terrain and weather data for a real forest demonstrate the viability of the integrated simulation environment for wildfire spread and containment.

Keywords: DEVS, dynamic structured DEVS (DSDEVS), fire spread, fire containment

1. Introduction

Wildfires play a very important role in the management of forests. These fires significantly influence forest management activities ranging from timber harvest scheduling to reforestation and thinning operations. Controlled prescribed fires help to maintain a manageable fuel loading for forests susceptible to destructive wildfires. However, wildfires have continued to threaten communities along the wildland urban interface (WUI) and often, destroy homes, wildlife and thousands of acres of prime forest land every year. This ecological problem raises significant concern that calls for extra attention requiring understanding the underlying causes, the effect of land management on fire ecology, wildfire risk, the dynamics of vegetation fuel, and how to reduce the likelihood of large scale fires. In the event of a wildfire, however, fire managers are faced with the difficult task of decision-making under uncertainty for the optimal allocation of the limited fire suppression and containment resources to effectively control the fire. Therefore, real-time decision support systems that integrate wildfire growth simulation and operations research models for decision-making under uncertainty should be developed. Such systems would assist fire managers at the tactical level to effectively bring under control potentially catastrophic wildfires, and allow for timely warning and well-coordinated evacuation plans.

In the US it is estimated that more than 11,000 communities adjacent to federal lands are at risk from wildfires [38]. Human-caused wildfires may be prevented through education and patrol, but nature-caused wildfires cannot. More than 77,000 wildfires were reported and more than 6 million acres were burned in the US alone [43] in 2004. About a billion dollars is spent annually on wildfire suppression and containment [43]. Throughout the US, personnel, equipment and financial resources are tremendously strained in wildfire suppression and containment. In fact, thousands of firefighters and support staff from both state and local agencies work in dangerous conditions in order to preserve forestry resources and protect human habitat and lives. This highlights the need for more effective and dependable tools for wildfire management.

Motivated by the above-outlined factors, an integrated simulation model for surface wildfire spread and containment called DEVS-FIRE is proposed. DEVS-FIRE is based on the discrete event system specification (DEVS) [47][48] and a wildfire spread model by [27], which focused on the principles of simulating wildfire behavior in DEVS. Since the work of [27] significant progress has been made to improve the fidelity and performance of DEVS wildfire simulations. These aspects include a new fire spread decomposition scheme [30], multi-resolution simulation [19], a hybrid agent-cellular space approach for fire containment simulation [17], dynamic structure (DS) modeling, and using real geographical information system (GIS) data. The incorporation of GIS technology has made it possible to develop detailed fire behavior predictions for numerous scenarios.

The contributions of this paper include a new integrated simulation environment for wildfire behavior and containment, and new results for simulated wildfires in a real forest using high resolution GIS terrain and fuel data, and real weather data. The rest of the paper is organized as follows. The next section gives a review of related work and Section 3 describes the new DEVS-FIRE model. Computer simulation and model validation results are reported in Section 4. The paper ends with a discussion and concluding remarks in Section 5. A brief overview of wildfire behavior basics is provided in the Appendix for the reader not familiar with the subject.

2. Related Work

DEVS is a sound formal modeling and simulation (M&S) framework based on generic dynamical systems concepts [44][47] and has been applied to both continuous and discrete systems. It has been an emerging paradigm for modeling complex adaptive systems [49] such as those arising in wildfire [27], distributed supply chain [20][46] and dynamic model reconfiguration and simulation control for the department of defense (DoD) design process [22]. DEVS has now become a practical simulation tool in a variety of implementations. For example, DEVSJAVA [48], an object-oriented Java M&S environment based on the parallel DEVS formalism [47], allows for quick development of reusable models and simulations. DEVS is also the basis for DEVS/HLA [48], a High Level Architecture (HLA)-compliant distributed M&S environment formed by mapping the DEVS-C++ system [45] to the HLA Runtime Infrastructure. The use of object-oriented technologies such as DEVS to build collaborative applications seem promising for decision support systems such as those for wildfire management.

Two of the widely distributed and accepted fire behavior predictive models are FARSITE [12] and BehavePlus [5][6]. Both models are used by fire behavior analysts from several wildfire agencies and are designed for use by trained wildland fire managers familiar with fuels, weather, topography, and wildfire situations. The fundamental difference between FARSITE and BehavePlus is in the way fire growth is modeled. FARSITE is based on Huygens' principle of wave propagation [[3], where fire growth is simulated as a two-dimensional elliptical wave [33] using spatial data from a GIS. In this approach the fire-front is projected over a finite time-step using fire behavior at discrete points along the fire's edge. Local raster information on fuels, topography and weather is used to compute a one-dimensional fire spread (speed and direction) for each point using the Rothermel [34] surface fire spread model. A two-dimensional fire growth is produced by aggregating all points around the fire perimeter. BehavePlus also uses the Rothermel model to compute the spread rate of the head fire. However, this value is used along with elapsed time to determine the size of an elliptically shaped fire [2][4]. This is also the approach followed in DEVS-FIRE. The rate of spread prediction using the Rothermel model, however, assumes that the weather, terrain and fuels remain uniform for the elapsed time. A more recent wildfire simulator, HFire [23], is a raster-based model for fire behavior through Southern California chaparral and also uses Rothermel's model.

The conceptual basis for a cellular discrete event hierarchical modular fire spread model using DEVS was introduced and illustrated by [39]. Discrete event models can take advantage of the heterogeneity of fire spread for faster simulations. More recently [27] developed a cellular DEVS fire spread and suppression model following along the line of work of [39]. Their model incorporates control response measures [29] and represents an advance toward developing a real-time decision support cellular simulation system for fire spread prediction and the effects of suppression attempts. The paper by [28] gives a formal expression of the forest cell model in parallel DEVS [47] and Timed Cell-DEVS [41] formalisms. The use of Timed Cell-DEVS in a simple rule-based cellular surface fire spread model was demonstrated by [1]. Cell-DEVS was also used to develop a physical model of fire spread in [25]. This model uses heat transfer partial differential equations to compute fire spread in each cell. In [26] the authors qualitatively compare DEVS and Cell-DEVS simulation results against controlled laboratory experiments which allowed them to validate both simulation models of fire spread. These authors were able to demonstrate how these techniques can improve the definition of fire models.

A DEVS hybrid agent-cellular space modeling approach for fire spread and suppression simulation was proposed by [17]. Their approach allows for simulating firefighting 'agents' with the ability to move within the cell space. The issue of two-dimensional fire spread decomposition in cellular DEVS models based on the Rothermel fire spread model is presented in [30]. Dynamic multi-resolution in cellular space modeling for forest fire simulation is considered in [19] for cases where fuel and spatial terrain data with different resolutions is available. This allows for comparing the accuracy of simulation results based on input data with different resolutions.

An important factor in cellular discrete event simulation models is whether all cells in the cell space are created at the beginning of the simulation or are created during simulation as needed using some DS approach. DS refers to the ability of a system to dynamically change its structure according to different situations. It provides a M&S environment with powerful modeling capability and the flexibility to simulate and analyze complex systems. In particular, DS makes it possible to load only a sub-set of system's components for simulation. This is especially useful in large scale wildfire simulations requiring large numbers of cells in the cell space. Previous work on DS has established a theoretical background and developed formalisms [7][37]. The work of [8] applied the DSDEVS to an example of fire spread simulation. A recently DS capability implemented in the DEVSJAVA environment [18] supports the wildfire spread and containment simulation model presented in this paper.

3. The DEVS-FIRE model

The DEVS-FIRE model provides an integrated M&S environment for both wildfire behavior and firefighting. This section describes DEVS-FIRE providing details on the overall system architecture (Section 3.1), wildfire behavior cellular space model (Section 3.2), DS cell space model (Section 3.3), and fire suppression and containment (Section 3.4).

3.1 System Architecture

The overall system architecture of DEVS-FIRE is shown in Figure 1. At the heart of the system is the DEVS cellular space fire spread model, which uses GIS terrain data, fuel model data, and weather data, through a Fuel, Terrain and Weather Data Interface layer. This allows each forest cell to be initialized with its fuel and terrain data, and to be updated with the weather data in real time. When a cell is ignited, Rothermel's mathematical model (Behave Model) is used to calculate the fire spread within the cell. To simulate fire containment, DEVS-FIRE uses an agent-based approach whereby the Firefighting Agent Model is used to model 'agents' representing different firefighting resources. The Firefighting Agent Model works together with the DEVS cellular space fire spread model to simulate both wildfire spread and firefighting scenarios. The deployment of firefighting agents is guided by a Stochastic Optimization Model [31], which takes the output from the wildfire spread simulation (burned area and fire perimeter predictions at given time steps) and Firefighting Resource Characteristics (e.g. type, arrival time to the fire location, production rate, rental cost and operating cost) to compute the optimal number of resources to dispatch to the wildfire to contain it as quickly as possible at minimal cost. In Figure 1 the Stochastic Optimization Model, Firefighting Resource Data Interface, and Firefighting Resource Characteristics components have dashed lines to indicate that these three components are not yet fully integrated into DEVS-FIRE. The Visualization component displays the dynamics of fire spread as well as that of firefighting agents.

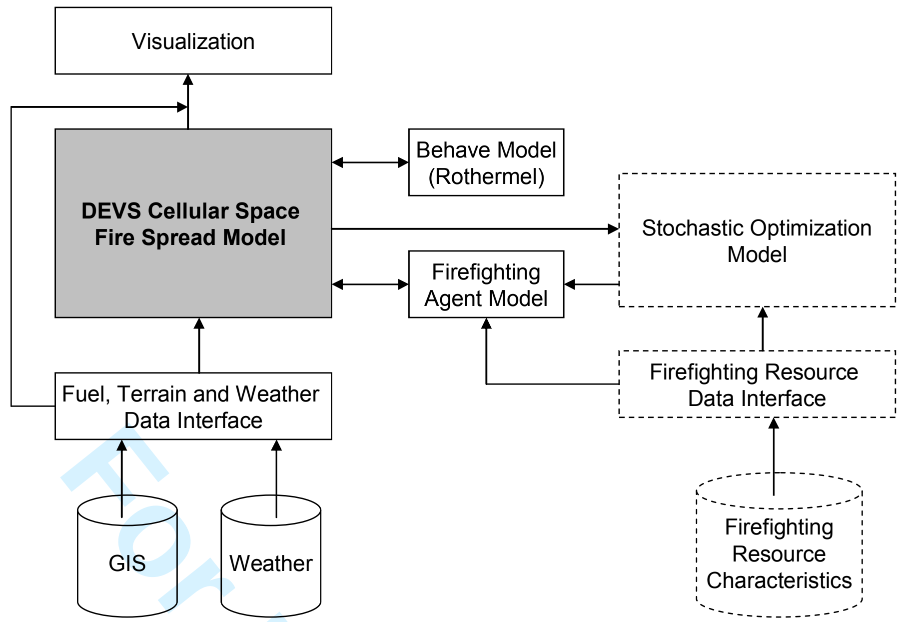


Figure 1. Overall system architecture of DEVS-FIRE

3.2 Cellular Space Model

In DEVS-FIRE the forest is represented as a two-dimensional cell space of rectangular cells whose dimensions depend on the resolution of the GIS fuel and terrain data. The cell space comprises individual forest cells with the fuel, terrain, and weather conditions assumed to be uniform within the cell. Each cell is represented as a DEVS atomic model in the simulation and performs its local computation of the rate of fire spread and direction based on its fuel, terrain, and prevailing weather conditions. DEVS allows for representing the forest cell as an atomic model with input and output ports between neighbor cells for exchanging messages. Consequently, the forest cell space is a coupled model composed of a number of coupled forest cell models. Fire spread across the cell space is enabled via message exchange between neighbor cells. The static grid cells representing space are external to the simulation and represent fuel and terrain conditions and fire location, while the forest cell models can be dynamically created in the simulation at runtime. Unlike in the previous DEVS wildfire spread model in [27], we follow a dynamic structure approach and allow cells to be dynamically created and deleted as needed at runtime. The burning process occurs in these cells and is computed and dynamically mapped at event instants into the static structure.

In DEVS-FIRE the behavior of a burning cell is influenced not only by external inputs from neighboring cells, but also dynamic changes in weather conditions and firefighting effects. Wind speed and wind direction are global external inputs to the cell space. Therefore, any changes in these variables are dynamically passed on to all the cells in the cell space. DEVS-FIRE allows for stochastic simulation by incorporating uncertainty in the model critical variables such as wind speed and direction. The variables, if not known with certainty, can be sampled from appropriate probability distributions. Consequently this allows for making several runs of the simulation with same initial input conditions but different scenario results. The scenario results include predictions of fire perimeter and area burned at given time steps, which are input for the stochastic optimization model for optimal firefighting resource dispatch for wildfire containment.

3.2.1 Cell States and State Transitions

The abstraction from the actual forest cell to an atomic forest cell model in DEVS permits this atomic cell model to be in only one of the following eight states at any time: *unburned*, *burning*, *burned*, *unburned-wet*, *burning-wet*, *burned-wet, unburned-attack*, and *unburnable*. Each cell is initialized in the *unburned* state (passive state) with its fuel and terrain parameters mapped from the forest cell weather and GIS fuel and terrain data. The weather data are assumed to be dynamically obtained from a weather station nearest to the fire location. The state transition diagram is given in Figure 2. A forest cell that transitions into an absorbing state (*unburned-wet*, *burned*, *burned-wet*, and *unburnable*) remains in that state for the duration of the simulation. A forest cell remains in the initial *unburned* state unless it is either ignited or affected by firefighting efforts. It transitions to the *burning* state if it receives a message from the Igniter and its *fireline intensity* [10] is above a threshold value set for the simulation. The cell transitions to *unburnable* state from *unburned* state if it receives indirect firefighting efforts. If in the *burning* state, the cell transitions to the *burned* state immediately after its 'burn time delay' has elapsed. The burn time delay is computed by the Behave Model (using Rothermel's model) and corresponds to the time it would take the fire to spread across the cell. Otherwise, the cell transitions to *burning-wet* if fire suppressant is introduced and either firefighting rule 1 or 2 (Section 3.4.1) is satisfied.

Once in the *burning-wet* state a cell remains in this state for a duration that is equal to the minimum of the 'burn time delay' and the 'direct-attack' time delay, which is a time duration determined by the Firefighting Agent Model (equal to the time for performing 'direct-attack' firefighting). The cell transitions to the *burned-wet* state if the burn time delay is less than the 'direct-attack' time delay. Otherwise, it transitions to the *unburnable-wet* state. Under 'indirect-attack' fire suppressant is introduced into the cell (or fuels removed) before it is ignited and firefighting rule 3 or 4 (Section 3.4.1) is satisfied. In this case, the cell transitions from the *unburned* state to *unburned-attack* and stays in this state for a time duration determined by the Firefighting Agent Model (equal to the time for performing 'indirect-attack') before transitioning to the *unburnable* state.

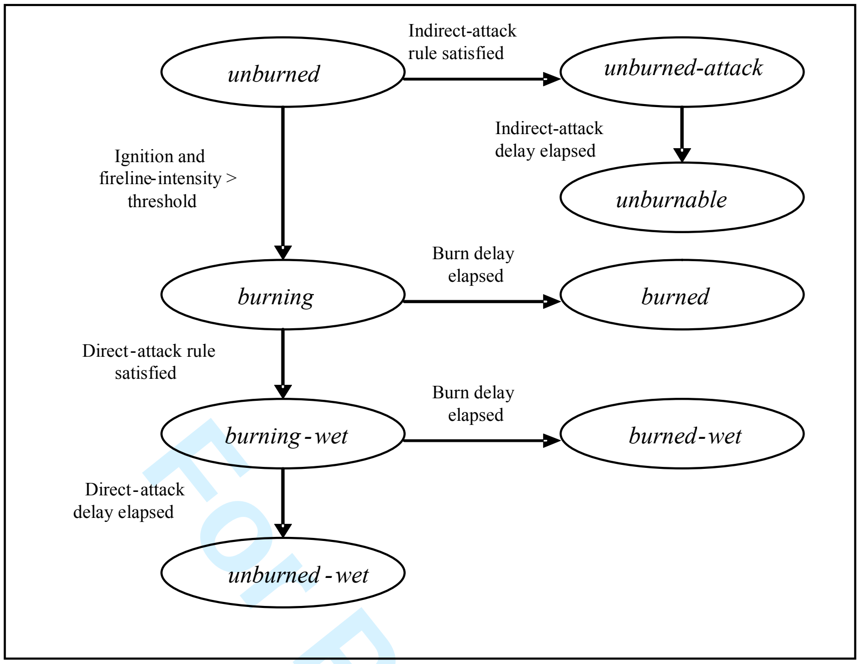


Figure 2. Forest cell state transitions

3.2.2 Fire Spread Decomposition Schemes

Currently DEVS-FIRE models fire spread in each cell according to Rothermel's [34] stationary model. Since this fire spread model is a one-dimensional semi-empirical model, and a propagation algorithm that uses maximum rate of spread and wind and slope factors is applied to obtain the second dimension. As in [27] each cell has fixed major spread directions N, NE, E, SE, S, SW, W, and NW. This restricts the number of directions for decomposing the maximum rate of spread obtained from Rothermel's model as is also done in [1][39], for example. As in FARSITE, BEHAVE and HFIRE, the DEVS-FIRE model also assumes elliptical fire shapes [2] in decomposing the cellular one-dimensional maximum rate of spread and direction from Rothermel's mathematical model to achieve two-dimensional spread.

In DEVS-FIRE three decomposition schemes are considered: center-to-center, center-to-border and border-to-border. Center-to-center assumes fire spreading from the center of the cell to the center of the neighbor cell, while center-to-border assumes fire spreading from the center of the cell to its border. Border-to-border assumes fire spreading across the cell from border-to-border. In these decomposition schemes both head fire and backfire are assumed to travel same distances. Therefore, the three decomposition schemes can result in apparent "faster" fire spread across the cell space and "thick" fire-fronts if proper care is not taken. For example, under the center-to-center scheme one needs to consider terrain conditions between cell centers and avoid doubly computing spread in a given direction. Further details on the decomposition schemes are given in [30].

3.3 Dynamic Structure (DS) Cell Space Model

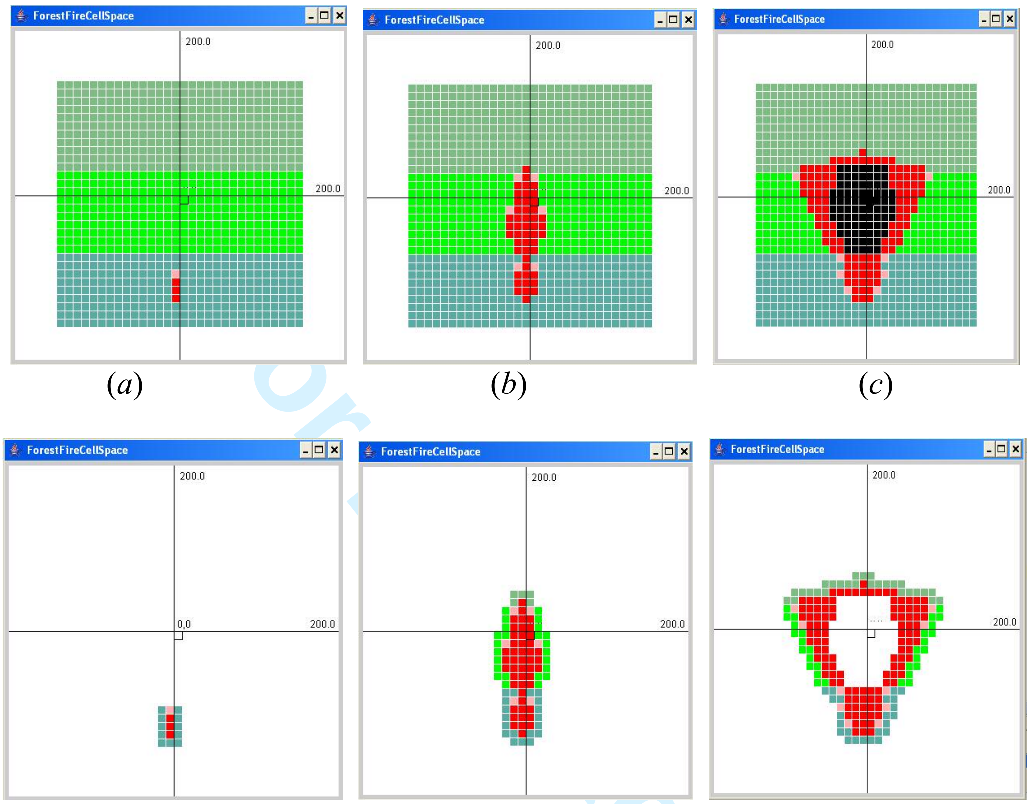
In the cellular space model presented in the previous section a large number of cells are needed to simulate a realistic size wildfire scenario. For example, the experiments described in Section 4 are for a 0.5 $km \times 0.5\ km$ (kilometers) area with total 40,000 (200×200) cells at resolution size of 2.5 $m \times 2.5\ m$. Simulating a large scale wildfire or multiple wildfires therefore requires a large number of cells. However, the large number of cells in a simulation poses several issues from the simulation performance aspect. Thus as an alternative to the standard implementation of the cellular space fire spread model, an approach of DS modeling and simulation is adopted in DEVS-FIRE. Different from the non-DS implementation that creates and loads all the cells at the beginning of a simulation run, the DS implementation starts with only the active cells that are ignited. As the simulation proceeds, other forest cells are dynamically created and added into the cell space when needed, that is, when they are about to catch fire. Meanwhile, when a forest cell is not needed, that is, after transitioning from an active state (*burning*, *burning-wet*, *unburned-attack*) to an absorbing state (*burned*, *burned-wet*, *unburnable*, *unburned-wet*), it is removed from the cell space. As a result, the dynamical structure implementation keeps only the forest cells along the fire-front in the cell space during a simulation.

Implementing the DS is motivated by several practical reasons related to simulation performance. First, running simulation with all the cells requires a large amount of memory for large scale cellular space models. In the forest fire spreading model, each cell is a complex atomic model that has its own attributes (internal variables), data structures (e.g., to keep track of fire progress along the eight directions), and behavior (specified by the state transition functions). Thus each cell occupies considerable amount of memory space. An estimation of memory usage based on the current implementation shows that each cell needs about 35 KB memory space [16]. For a large scale fire spread model such as the one with 200×200 cells, 1.4GB memory is required in order to run the simulation. We note that this implementation could be optimized to make it more memory efficient. However the fundamental issue is that the larger the cell space, the larger the memory needed to load all the cells into the simulation. Second, since the non-DS implementation loads all the cells at the beginning of a simulation run, the initialization time of the simulation is much longer as compared to that of the DS implementation. In our experience, it takes several minutes to initialize a simulation with 100×100 cells. This is undesirable when compared to the DS implementation which can start the simulation in a few seconds. Third, from the simulation speed point of view, DS implementation brings some computational overhead by dynamically adding/deleting cells at runtime. However, such overhead is not critical for the application of wildfire spread simulation since the number of active cells is typically very small when compared to the total number of cells in the cell space.

To implement the DS model, we took advantage of DEVSJAVA's variable structure modeling capability that allows dynamically adding and removing models at the same level of model hierarchy [18]. Specifically, a *DynamicCell-SpaceManager* atomic model was developed. This model is a sub-component of the cell space model and is responsible for dynamically adding and removing forest cells when needed. To make the DS modeling work, a forest cell model has two extra output ports *outBurning* and *outBurned* defined. These two ports are coupled to the *DynamicCellSpaceManager*'s two input ports, *inBurning* and *inBurned*, respectively. When a forest cell is ignited it sends out an "adding" message via its *outBurning* port to *DynamicCellSpaceManager*'s *inBurning* port. In response to this message, the *DynamicCellSpaceManager* dynamically creates and adds the requesting cell's neighboring cells as well as their neighbor-to-neighbor couplings. Similarly, whenever a cell is about to transition to an absorbing state, it sends out a "delete" message via its *outBurned* port to *DynamicCellSpaceManager*'s *inBurned* port. This triggers the later to remove the requesting cell from the cell space. This process of adding and removing the cells from cell space continues until the simulation ends. The pseudo code of the *DynamicCellSpaceManager*'s external transition function that is in charge of adding/removing forest cells is shown below.

```
if (messageOnPort( "inBurning")) {
    get the ID of the requesting cell;
    for (all the neighboring cells){
        if (cell has not been loaded)
            create the cell;
            addModel(the created cell);
            addcouplings;
        }
    }
}
else if (messageOnPort("inBurned")) {
    get the ID of the requesting cell;
    removeModel(the requesting cell);
}
```

Figure 3 gives example results of a comparison between the non-DS (plates *a*, *b*, and *c*) and DS (plates *d*, *e*, and *f*) implementation for the same wildfire spread model at three different stages. The figures are better viewed in color. For the figures at the top row, the red cells are burning; the black cells are burned out; the pink cells are just ignited and transitioning to the burning state; all other cells are unburned with the different colors representing different fuel models. In the bottom row figures, the white spaces indicate the cells (that are either unburned or burned out) that are not loaded as part of the model. All other cell colors have the same meaning as described above. This comparison shows two important features about the DS implementation. First, the DS and non-DS implementations lead to the same simulation results. This validates the correctness of the DS implementation. Second, the comparison clearly shows the difference between the two implementations. In the DS implementation, cells are dynamically added when they are about to be ignited by their neighbors, and removed when they are burned out. However, in the non-DS implementation, all cells are loaded from the beginning and kept throughout the simulation. The standard DEVS coordinator was used as the simulation engine. In the non-DS case the execution times from the beginning of the simulation corresponding to the plates *a*, *b*, and *c* in Figure 3 are 3.86*s*, 6.27*s*, and 19.3*s*, respectively. For the DS case, the execution times corresponding to the plates *d*, *e*, and *f* are 0.23*s*, 1.58*s*, and 13.2*s*, respectively. This shows significant improvement in execution times with the DS implementation. The ratios of the number of active cells to the total cells in the cell space for the three snapshots are 0.003, 0.052 and 0.120, respectively.



(*a*) (*b*) (*c*)

(*d*) (*e*) (*f*)

Figure 3. Comparing the non-DS and DS implementation

A detailed performance measurement and analysis of DS modeling for forest fire simulation is not the focus of this paper and can be found in [36]. However, it is worthy to point out two things that are related to the simulation speed of DS modeling. First, as measured in [36], the DS implementation introduces an overhead of dynamically adding/removing models, which is proportional to the number of models that need to be added/removed in every simulation step. Thus for applications where every simulation step has a high demand of adding/removing models, the overhead will become significant and may even slow down the simulation as compared to a non-DS implementation. However, for the wildfire spread simulation, a relatively small number of cells (compared to the total number of cells in the cell space) are active and need to be added/removed in every simulation step. Consequently, the overhead is relatively small. Second, in a non-DS implementation, the large number of cells poses an algorithmic challenge of how to efficiently find the *imminent cells* that have the smallest next event time in every simulation step. The standard DEVS coordinator is inefficient in this manner because it scans all the cells in order to find the imminent cells, which has the computation complexity of $\Omega(N)$ where N is the total number cells in the cell space. With the DS implementation, however, only a small portion of the cells that are active are kept in memory and that problem does not arise. We note that without using DS, the above problem can be solved by developing more advanced simulation engines that use advanced data structures to keep track of the imminent cells (see e.g., [16]).

We note that the current DS implementation uses a central *cellSpaceManager*, which may cause a performance bottleneck. A different design could be implemented in a distributed manager, whereby each cell is responsible for adding its neighbors or removing itself dynamically as the simulation proceeds. Also note that the DS implementation is specifically tailored to the wildfire spread simulation.

3.4 Fire Suppression and Containment

Besides wildfire behavior simulation, DEVS-FIRE also supports fire suppression simulation. The interaction between firefighting agents and wildfire behavior models allows for studying the effectiveness of different firefighting strategies and different firefighting resource dispatch plans for given wildfire behavior scenarios. As pointed out earlier, the wildfire suppression simulation takes the output from the Stochastic Optimization Model regarding the optimal firefighting resources to dispatch to a wildfire as input. Integration of wildfire suppression simulation and stochastic optimization is still under development. This section discusses the system design to support agent-based firefighting simulation in DEVS-FIRE.

3.4.1 Agent-Based Firefighting Simulation in DEVS-FIRE

To support firefighting simulation based on the wildfire spread models described above, DEVS-FIRE adopts a hybrid agent-cellular space modeling approach [17], where cellular space models are used to model the dynamics of wildfire spread, and agent models are used to model the firefighting resources such as fire-fighters and air-tankers. This hybrid agent-cellular space modeling approach separates the design concerns of wildfire spread and firefighting. The cellular space model is responsible for capturing the dynamics of wildfire spread while the agent model is responsible for modeling the firefighting actions based on firefighting rules and tactics. The loose coupling between the firefighting and wildfire spread models makes it easy to evolve each one independently. For example, new firefighting tactics such as direct (head and tail) attack, parallel attack, and indirect attack (see e.g. [14]) can be added into the agent models without affecting the wildfire spread model.

Figure 4 illustrates the model structure that integrates agents and cellular space models for wildfire spread and suppression simulation. Only one agent is shown for illustration purpose. However, the figure can be expanded to situations with multiple agents. As shown in the figure, there are four loosely coupled components: *Forest Cell Space Model*, *Agent* model, *couplingManager* model, and *fireManager* model, which are involved in simulating wildfire suppression. In general, an agent model moves in the cell space and influences the corresponding cells' wildfire behavior. To carry out firefighting actions, an agent needs to know the fire spread conditions in its environment (the cellular space) and then take actions to affect the environment. To support this interaction between an agent and its environment, couplings are added between the agent and the corresponding cell where the agent locates. These couplings are dynamically added/removed (using the DS approach) during the simulation when the agent changes its location from one cell to another.

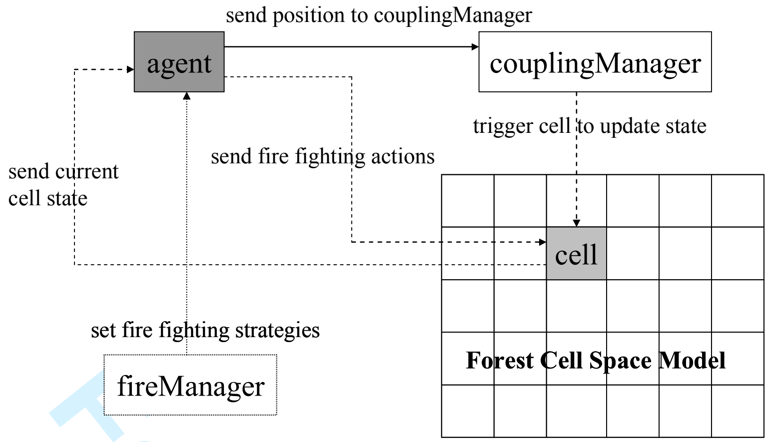


Figure 4. Architecture for hybrid agent and cellular space modeling

Minor changes need to be made to the *Forest Cell Space Model* to support the interaction between firefighting agents and forest cells in firefighting simulation. Specifically, a new *queryState* port is added for each forest cell. Whenever a cell receives a message on this port, it sends out a message that contains its current state. Also, a cell will send out its state whenever it transitions to a new state. This allows the agents coupled to this cell to know the current state of the cell.

The *Agent* model is used to model firefighting resources. An agent can move in the cellular space with a certain speed (e.g., the production speed of suppressing a wildfire) and along a certain direction (e.g., according to a planned route as in indirect attack). During the movement, an agent keeps track of its own position and constantly sends its position to the *couplingManager*. Meanwhile, it continuously monitors the condition (state) of its corresponding cell and, if necessary, takes fire suppression actions based on certain wildfire suppression rules, such as the ones used in [29] and restated below. These rules are adapted from the work of [5] and [35]. The first two rules allow for *direct attack*, which in our context means that firefighting efforts are directed on *burning* forest cells. The last two rules constitute *indirect attack* and refer to firefighting efforts directed on *unburned* forest cells ahead of the fire-front that have not yet caught fire.

Rule 1. *If (flame length < 1.2 m) fires can generally be attacked at the head or flanks of the fire by persons using hand tools*

Rule 2. *If (1.2 m ≤ flame length < 2.4 m) fires are too intense for direct attack at the head of the fire by personnel with hand tools but equipment such as bulldozers and retardant aircraft may be effective.*

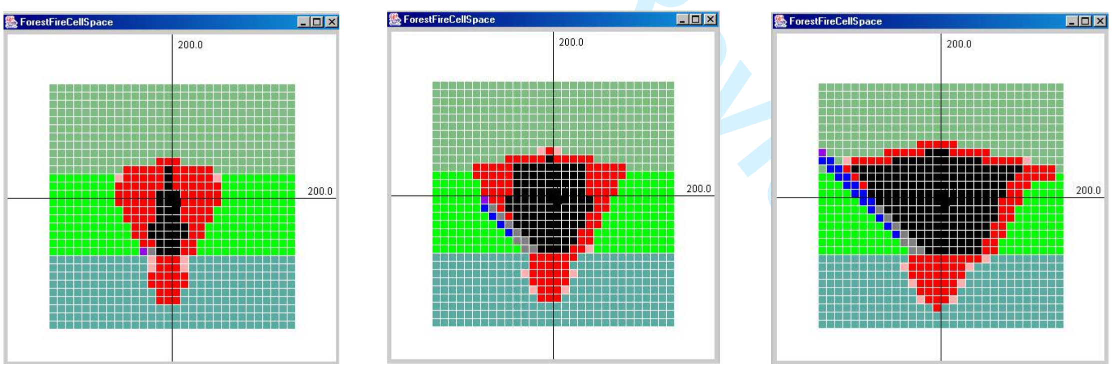
Rule 3. *If (2.4 m ≤ flame length < 3.4 m) control effort of the fire will probably be effective. Indirect attack is the only means of suppression.*

Rule 4. *If (flame length ≥ 3.4 m) control efforts at the head of the fire are ineffective by any known means of suppression. Indirect attack may be the only means to slow the spread of the fire in certain directions.*

To support the interactions between an agent and its location cell, the *couplingManager* model takes care of the coupling changes when an agent moves in the cellular space. It receives messages that contain the agent's (new) positions (x, y) from the agent. This message triggers the *couplingManager* to find the cell where the agent locates. If the cell ID has changed, couplings between the agent and the old cell will be removed and couplings between the agent and the new cell will be added. Furthermore, a coupling is added from the *couplingManager* to the new cell. This coupling allows the *couplingManager* to inform the new cell to send out its current state. Thus whenever an agent is coupled to a new cell, it will receive a message from the cell that contains the cell's current state. The couplings that are dynamically added/removed are represented in dashed lines in Figure 4. To give an example, when the agent changes its location from an old cell to a new cell, the *couplingManager* executes the following code fragment to remove a coupling from an agent to the old cell and to add a coupling from the agent to the new cell. In this sample code, the *ffAction* is the agent's output port which sends out firefighting actions (commands), and the *inFireFight* port is a cell's input port that receives firefighting actions.

```
removeCoupling( agent , "ffAction", oldCell, "inFireFight");
addCoupling(agent , "ffAction", newCell, "inFireFight");
```

The fourth part of this architecture concerns the *fireManager* that is part of the Stochastic Optimization Model represented by the dotted box in Figure 1. During the process of wildfire suppression, an agent may receive high-level commands from the *fireManager*, whose role is to allocate firefighting resources and set firefighting strategies from a global point of view.



(*a*) (*b*) (*c*)

Figure 5: A firefighting agent in action

Figure 5 shows an example of agent-based firefighting with one agent at three different stages of wildfire spread and containment. The pictures are better viewed in color. In Figure 5(*a*), the agent (in purple color) is deployed to a forest cell at the fire-front. This agent is pre-defined to move northwest (at a speed of 5 *m/s*) and to take firefighting actions, i.e., adding water to the cells, along the path. A random number is used to simulate the time for the agent to carry out the firefighting action. As the result of firefighting efforts, a *burning* cell transitions to the *burning-wet* (grey color) state while an unburned cell transitions to *unburned-wet* (blue color) state. This is displayed by

Figure 5(*b*), which shows that the agent has moved a distance along a northwest direction and has succeeded in making the corresponding forest cells wet. Because of this, the fire is not able to spread along the southwest direction across the wet cells. This is further illustrated by Figure 5(*c*), where the agent essentially creates a strip of "safe zone" to prevent the fire from spreading across it. This simple example demonstrates that the firefighting agent can work with forest cells for simulating the dynamics of both wildfire spread and containment. It builds the ground to develop more advanced and more realistic wildfire suppression simulations.

3.4.2. Interface with Stochastic Optimization

Fire managers are faced with the difficult task of making strategic and tactical decisions under uncertainty regarding the deploying of firefighting resources within a limited budget. The main source of uncertainty is in the evolution of the wildfire. The strategic decisions include long-term plans for the attack-bases and associated firefighting resource allocation. Tactical decisions involve short-term operations and scheduling of the resources with respect to actual wildfire occurrence. DEVS-FIRE is designed to provide stochastic information about wildfire growth that is necessary input to a tactical stochastic optimization decision-making model for determining the optimal mix of the firefighting resources to deploy to contain a wildfire. Such information includes scenario predictions of the fire perimeter and burned area at given time periods in the future from the time the fire is reported. To this end, [31] have proposed a stochastic programming [9] model to interface with a surface fire simulator such as DEVS-FIRE. Their model is a two-stage stochastic program based on the widely used cost plus net value change (C+NVC) model for wildfire economics [15].

The objective function of the stochastic programming model is to minimize the expected total cost of wildfire which is the pre-suppression costs plus the expected suppression costs and NVC. NVC is the dollar value associated with the net damage to a given area of the forest due to the fires in a given time period. The model assumes that that if the total line production of the fire-fighting resources exceeds the total fire perimeter then the fire is contained. Therefore, data on the available firefighting resources with their characteristics is also input to the model. The firefighting resource characteristics include fireline production rate, arrival time to the fire, rental cost and operation cost. The two-stage model selects resources to dispatch to the wildfire in the first-stage. In the second-stage, given the resources to dispatch and a collection of wildfire growth scenarios (fire perimeter and burned area at given future time periods), the model makes corrective (recourse) actions on actual fire containment. Note that because of budgetary and resource constraints, it is imperative to determine whether the wildfire can be contained or not. The model can be solved to determine whether or not the fire can be contained for a given budget and firefighting resources. If the fire can be contained, the model then identifies the optimal mix of resources to dispatch with the minimum expected total cost.

4.0 Computational Simulation Experiments and Validation

Several computational experiments were conducted with DEVS-FIRE to simulate wildfires occurring in a real forest located in the Huntsville area, Texas, USA. The aim of the experiments were threefold: (1) to test and validate DEVS-FIRE wildfire spread predictions under different fuel, terrain and weather conditions based on a validated wildfire spread model from the literature, (2) to demonstrate the ability of DEVS-FIRE in predicting fire spread using fuel, terrain and weather data for a real forest, (3) to demonstrate the ability of DEVS-FIRE with using fuel data of different types and different resolutions. The simulations were conducted on a Toshiba laptop with Intel Celeron (M) 1.6GHZ processor, 1.2G memory, and Windows XP OS running DEVSJAVA version 3.0. Wildfire growth images were captured at preset simulation times and the burned area, fire-front perimeter size, and the ratio of the number of burning cells to the total number of cells in the cell space recorded.

Spatial fuel and terrain data for a study area of about half a kilometer in both length and breadth was provided to us by the Spatial Sciences Laboratory of the Department of Ecosystem Science and Management at Texas A&M University. A total of thirteen standard fuel models have been identified for the US [18], but only seven of these are available in the study area and they are as follows: Fuel model 1: Short grass (1 foot); Fuel model 2: Timber (grass and understory); Fuel model 4: Chaparral (6 feet); Fuel model 5: Brush (2 feet); Fuel model 7: Southern rough; Fuel model 8: Closed timber litter; and Fuel model 9: Hardwood litter. Fuel models 1 and 2 belong to the grass models, fuel models 4, 5 and 7 belong to the brush models, and fuel models 8 and 9 belong to the timber liter group. The terrain data for the study area was airborne LiDAR (Light Detection and Ranging) [40] raster-based GIS data with associated fuel model data of two types. The first type is one obtained by classifying a multispectral QuickBird (DigitalGlobe) image and the second is one obtained by classifying a LiDAR and Quickbird fused data set [24]. The LiDAR data were acquired during the leaf-off season for the Huntsville area, Texas, in March 2004 by M7 Visual Intelligence of Houston, Texas.

The LiDAR system (Leica-Geosystems ALS40) uses advanced technology in airborne positioning and orientation, enabling the collection of high-accuracy digital surface data. The horizontal and vertical accuracies with the LiDAR system for the data collection were 20-30 *cm* and 15 *cm*, respectively, with the system providing a 25 degree swath from nadir, with a cross-hatch grid of flight lines resulting in an average of 2.6 laser points per m^2. The point density translates into an average distance between laser points for the entire cloud of about 0.62 *m*. As described in [24], in processing the data, LiDAR height bins were generated as multiband images of 0.5 m height intervals and 2.5 $m \times 2.5\ m$ pixel dimensions, up to 2 *m* above ground. To map surface forest fuel models, the LiDAR height bins were stacked with a QuickBird image covering the same area and image processing techniques were applied to the fused dataset. For our experiments, we also obtained data processed for cell size resolutions of $5\ m \times 5\ m$. Weather data for a 24-hour period in March 2004 was available from a weather station in the study area.

4.1 Effect of Extreme Slope and Wind Conditions on Fire Spread

To accomplish the first objective of our experimental study we conducted an experiment to test DEVS-FIRE fire spread prediction under extreme slope and wind speed conditions.

Similar experiments where conducted in [27] for testing the center-to-center decomposition scheme. Here we test DEVS-FIRE predictions using the forward cell border-to-border fire spread decomposition scheme described in Section 3.2.2.

The first experiment was to study the effect of extreme terrain slope conditions on the rate of spread for wind speed arbitrarily fixed at 2.235 *m*/sec blowing up the slope. Three fuel models where arbitrarily chosen due to their differences in fuel loadings, fuel models 4, 7 and 11. As noted in [27], these fuels exhibited varied fire spread behaviors. The results of the experiment are given in Figure 6 and show an increase of the rate of spread with slope as expected. Fuel model 4 has a higher increase in the rate of spread followed by fuel model 7, which has a higher rate of spread than fuel model 11.

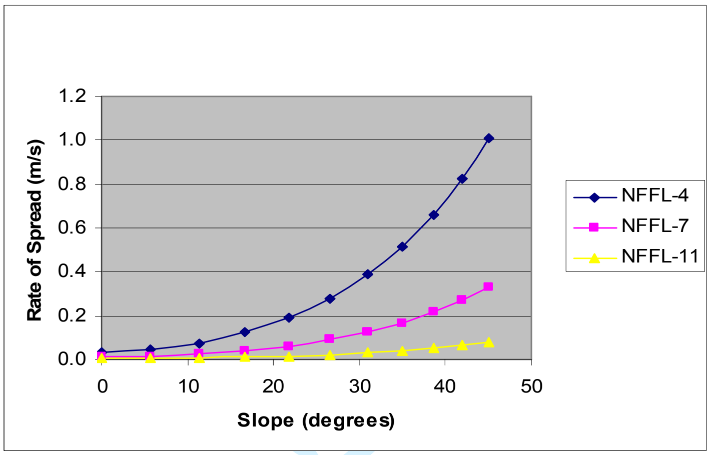


Figure 6. Rate of wildfire spread under extreme slope conditions

The second experiment was to study the effect of extreme wind conditions on fire spread in the three fuel models on flat terrain (slope fixed at 0 degrees). The wind speed was varied from 0 *m*/sec to 10 *m*/sec and the rate of spread recorded. The results are given in Figure 7 and show a steady increase of the rate of spread with wind speed. Again, fuel model 4 has a higher rate of increase in spread followed by fuel model 7 and 11, in that order. The results obtained for fuel model 4 agree within 10% of what is reported in [23] using HFIRE, which has been validated for fire spread in fuel model 4 as well as Ceanothus Chaparral.

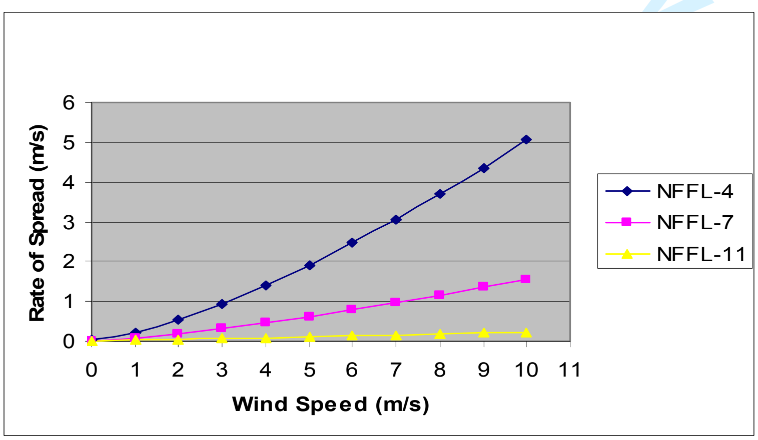


Figure 7. Rate of fire spread under extreme slope conditions

4.2 Fire Spreading under Different Fuel Model Type Data and Resolution

A set of experiments were performed to demonstrate the ability of DEVS-FIRE to predict fire spread using real fuel, terrain and weather data, with fuel data of different types. A wildfire burning in the study area was simulated using same LiDAR terrain data but with two the types of fuel data, QuickBird (DigitalGlobe) and LiDAR-QuickBird [24], respectively. Even though the experiments are based on the DS implementation, the simulation results are however displayed in the same way as in the non-DS implementation for clarity. The pictures are better viewed in color. The different shades of green in the pictures represent the different fuel models, with the lighter shades representing the lower numbered fuel models. Only about a quarter of the entire cell space is shown in the pictures based on the location of the wildfire to allow for smaller pictures.

Figure 8 shows the results of the simulation runs using QuickBird (DigitalGlobe) and LiDAR-QuickBird fuel model data with cell size cell-size 2.5 $m \times 2.5\ m$. In the figure, plates *a*, *b*, and *c* were captured at simulation next event times (tN) 300, 1500, and 2100, respectively. As can be seen in the figure, the fire is arbitrarily started from the center of the study area and spreads outward based on the fuel, terrain, and weather conditions. It can be seen that fire spreads much faster in the lighter shaded areas, which represent the grass fuel models. However, fire spread is significantly differently under the two fuel model data. This can be attributed to the inherent differences in the accuracy of the data. Fire spread is seen to be much faster with QuickBird fuel model data than with LiDAR-QuickBird data. It is also interesting to note how the fire spreads much faster in high-energy fuels, leaving patches of unburned areas as one would expect in a real wildfire. Next we simulated a wildfire burning in the study area using same fuel model data and terrain data but with increased resolution of cell size of 5.0 $m \times 5.0\ m$. The simulation results are given in Figure 9. Compared with Figure 8, fire spread under the two resolution data is very similar as can be seen by the shapes of the fire perimeter. However, the higher resolution data results provide more details on the fire-front location than the lower resolution data.

The ratio of the number of burning (active) cells to the total number of cells in the cell space or "active cells ratio", the outer fire perimeter, and burned area corresponding to each plate in Figures 8, and 9 are reported in Table 1. The ratios are useful in discrete event simulation in determining the efficiency of the simulation since they are a strong indication of the average number of *imminents*. The ratios are in fact very small as pointed out earlier, an indication that very few cells are actually burning (active) at any given time in relation to the total number of the cells in the cell space. Fire perimeter and area burned are seen to increase with time as expected. Also, the higher resolution data has generally more burning (active) cells than the lower resolution data, an indication that for a given area fire spreads across more cells in the higher resolution data than the lower resolution data. In this case we see that the results show larger ratios than for the lower resolution data.

2.5 m, QuickBird Fuel

2.5 m, LiDAR-QuickBird Fuel

(a) $tN = 300$

(b) $tN = 1500$

(c) $tN = 2100$

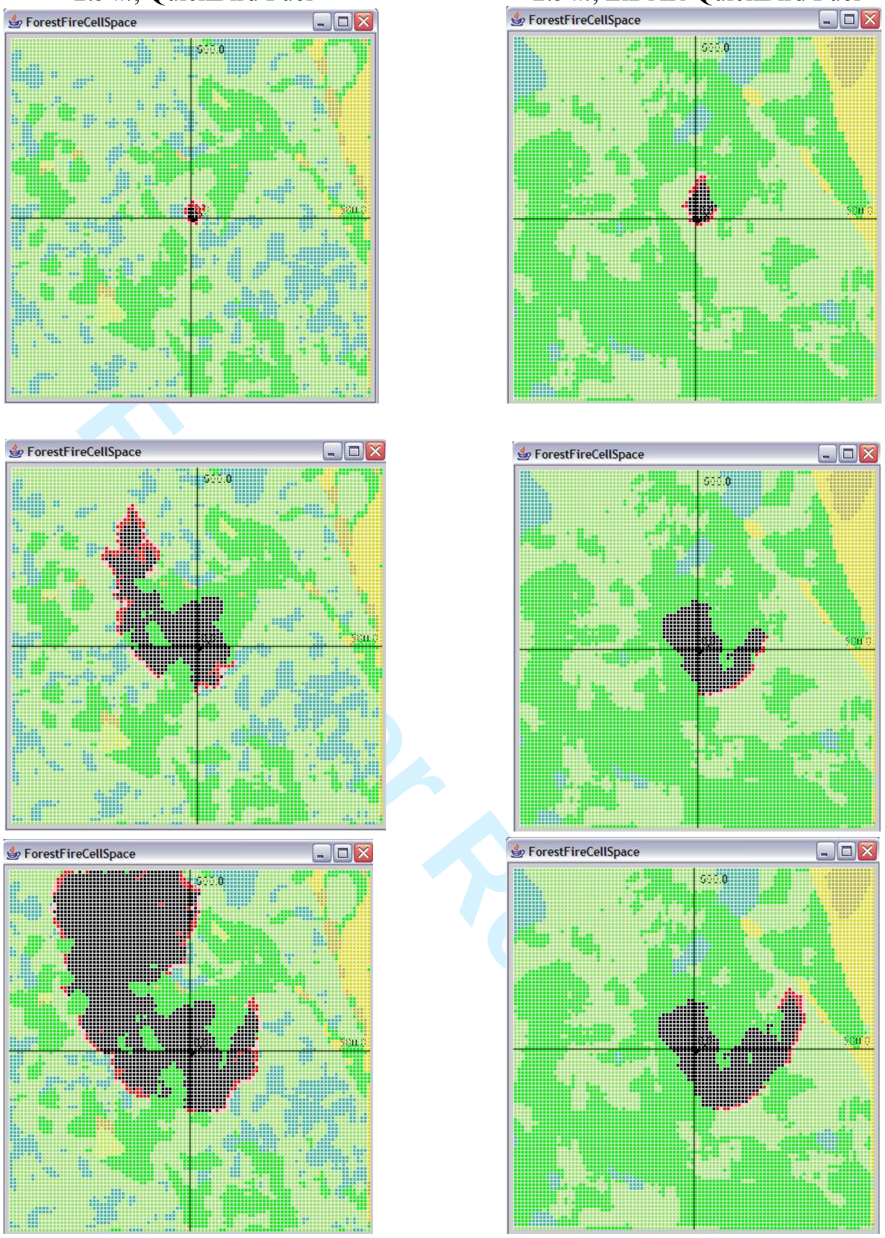
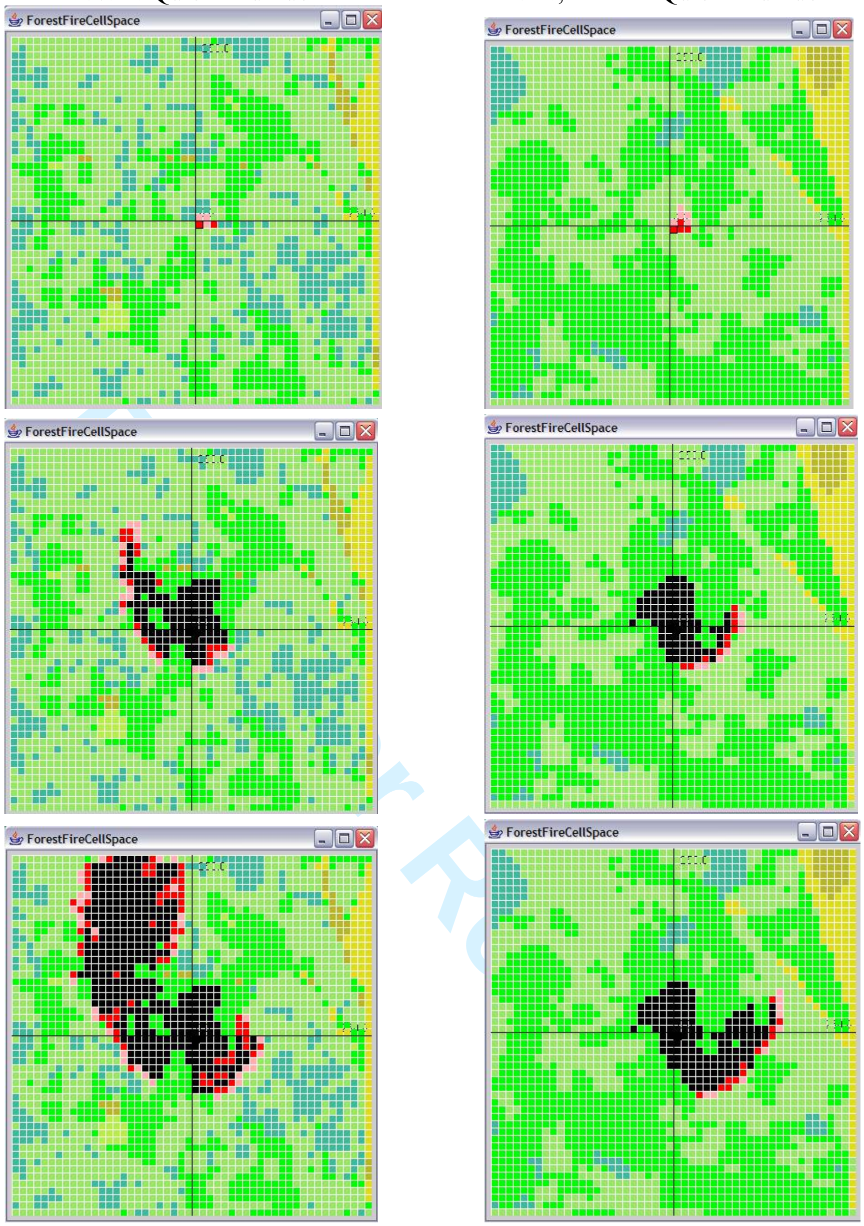


Figure 8. Fire spreading under different fuel model data with 2.5 m × 2.5 m resolution

5.0 *m* QuickBird Fuel 5 *m*, LiDAR-QuickBird Fuel



(a) $tN = 300$

(b) $tN = 1500$

(c) $tN = 2100$

Figure 9. Fire spreading under different fuel model data with 5 *m* × 5 *m* resolution

Table 1. Active cells ratio, perimeter and burned area for different fuel data

Figure 8	2.5 *m* QuickBird Fuel			2.5 *m* LiDAR-QuickBird Fuel		
Plate	Ratio	Perimeter (*m*)	Burned Area (m^2)	Ratio	Perimeter (*m*)	Burned Area (m^2)
a	0.0011	277.5	81.25	0.0014	400.0	518.75
b	0.0064	1787.5	5575.00	0.0011	337.5	3550.00
c	0.0094	2720.0	21287.50	0.0024	682.5	5868.75
Figure 9	5 *m* QuickBird Fuel			5 *m* LiDAR-QuickBird Fuel		
Plate	Ratio	Perimeter (*m*)	Burned Area (m^2)	Ratio	Perimeter (*m*)	Burned Area (m^2)
a	0.0006	90.0	0.0	0.0012	165.0	0.0
b	0.0059	875.0	3975.0	0.0021	315.0	3125.0
c	0.0211	3035.0	13875.0	0.0031	470.0	5075.0

5. Discussion and Conclusion

Simulating wildfire spread and containment remains a challenging problem due to the complexity of wildfire behavior. In this paper, a discrete event cellular space-based model for integrated surface wildfire spread and containment called DEVS-FIRE is presented. The cellular space model is based on the dynamic structured DEVS (DS-DEVS) and builds on a previous DEVS wildfire model. The new model allows for forest cells to be dynamically created and deleted from the cell space as needed, and incorporates real spatial fuels data, topographic data, and temporal weather data into the prediction of wildfire behavior across both time and space. DEVS-FIRE is designed to be integrated with a stochastic optimization model that uses the scenario results from the simulation to determine the optimal firefighting resources to dispatch to containment a wildfire as quickly as possible with minimal cost. Preliminary simulation results with fuel and terrain GIS data for a real forest demonstrate the viability of using DEVS-FIRE for wildfire spread prediction and containment.

The experiments demonstrate using DEVS-FIRE to run simulations with different types of GIS data with different spatial resolutions. The experiments also show that to simulate a wildfire in a real forest, the cellular space includes a large number of forest cells. Although the total number of cells is large, the percentage of active cells at any given time is very small as evidenced by the very small active cells ratios. The DSDEVS implementation takes advantage of this property and improves on the simulation performance for both the execution time and memory usage. Interested readers are referred to [36] for a comprehensive analysis of the performance gains and overheads introduced by dynamic structure implementation in DEVS.

From the experiments of using real GIS data, two observations are obvious. First, different GIS data gives significantly different fire spread results. Thus the 'right' GIS data is very important for a simulation to give precise fire spread predictions, which are critical in making wildfire containment decisions. Second, for the same type of GIS data, simulations using different resolutions result in similar fire shapes. However, a multi-resolution approach may be necessary to support simulations with different precisions and execution times by using different resolution data.

Future work along this line of research includes validation of DEVS-FIRE using historical wildfire data, incorporating other wildfire spread mathematical models different from Rothermel's model, developing more advanced fire suppression simulations with realistic tactics, and integrating stochastic optimization models for wildfire containment decision making under uncertainty. We believe that integrating wildfire spread predictions with operations research models would provide effective tools for both strategic and tactical wildfire management.

6. Appendix: Wildfire Behavior Basics

In this appendix we review the basics of wildfire behavior based on [32] to set the ground for the DEVS-FIRE model. The three important factors that influence wildfire behavior are vegetation, terrain and weather. Despite the fact that the influence of each factor on wildfire behavior is complex due to interactions between the factors, several generalizations have been made in the literature. In wildfire behavior literature vegetation is described by fuels, which refer to the composite of variables that describe the vegetation the fire is spreading through. A fuel description includes measurements of mass per unit area (load), energy per unit mass (heat content), surface-area-to-volume ratio, height, and moisture content. Terrain variables include slope and aspect. Slope is the inclination of a land surface relative to the horizontal, while aspect is the direction the surface is facing.

Fire spread can be described as the propagation of a flaming front that involves a series of ignitions whose heat brings successive stripes of fuel to the ignition temperature via a contagion process. This process is considered to be in steady-state for homogeneous fuels and unsteady-state for nonhomogeneous fuels [34]. Basically, energy from combusting fuel particles at the fire-front is transferred to unignited fuel particles ahead of the fire-front via the heat transfer mechanisms of radiation, convection and conduction [11].

To make accurate predictions of wildfire spread, accurate fuel, terrain and weather data are required. There are two approaches for predicting fire spread, the *physical* approach and the *empirical* approach. The physical approach considers fire spread as heat transfer between burning and unburned fuel using partial differential equations to solve for predicted fire spread under the assumption that all heat transfer involved in the combustion reaction satisfies the conservation of energy [42]. The empirical approach relies on statistical correlation between variables known to influence fire spread with field observations of rates of spread [34]. Therefore, this approach attempts to isolate and measure the effects of each variable using experimentation to develop equations for predicting fire spread.

In general, the rate of spread of a fire increases with the slope assuming all other conditions remain the same. This can be explained by the fact that as the slope increases, more fuels are exposed to the flame and the distance between the flame and unignited fuels ahead of the flame decreases. Consequently, more radiative heat energy reaches the fuels ahead of the flame resulting in faster heating of the fuel particles and ultimately, a higher rate of spread. Aspect dictates how much direct sunlight throughout the day the fuel receives, which in turn influences environmental conditions that affect the production of biomass, and hence the amount of available fuel. Note that aspect also affects the ambient fuel temperature. Therefore, fuels at slopes receiving more direct sunlight are generally at elevated temperature and may require less energy to be raised to their ignition temperature.

Unlike fuel and terrain, weather has a dynamic influence on wildfire behavior. The three components of weather that greatly influence fire spread are wind speed, wind direction and moisture content. Like slope, the rate of fire spread generally increases with wind speed. This effect can be attributed to the fact that wind induces a forward lean on the flame front in the prevailing direction of the wind resulting in decreased distance between the flame front and the unignited fuel particles. Wind also raises the rate of convective heat transfer between the heated air and the unignited fuel particles. Furthermore, as wind moves across the interior of the fuel bed, it increases the loss of moisture in the fuel particles by evaporation, decreasing the energy required for ignition.

The moisture content of the fuels dynamically changes with the weather. In living plants the fuel moisture content varies on a seasonal basis as the plant grows while in dead biomass it varies diurnally with the ambient temperature and humidity. Fuels with high moisture content retard the rate of fire spread due to the additional energy needed to vaporize the moisture and bring the fuel particles to ignition temperature. *Fireline intensity*, the product of the available heat of combustion per unit area of ground and the rate of spread of the fire [10], can be used to determine if a fuel is burnable or not. For example, [21] have determined fireline intensity threshold values for wildfire regimes in the Sierra Nevada.

7. Acknowledgements

This research is supported by a grant (CNS-0540000) from the National Science Foundation under the Dynamic Data Driven Applications Systems (DDDAS) program. The authors wish to thank Sorin Popescu and Muge Mutlu of the Spatial Sciences Laboratory at Texas A&M University for their insightful discussions and for providing the fuel, terrain, and weather data used in the experiments. The authors also thank the anonymous reviewers for their valuable comments that helped improve the presentation of this paper.

8.0 References

[1] Ameghino J., A. Troccoli, G. Wainer. 2001. Models of complex physical systems using Cell-DEVS. *Proceedings of 34th Annual Simulation Symposium.* Seattle, WA. U.S.A.

[2] Alexander, M.E. 1985. Estimating the length-to-breadth ratio of elliptical forest fire patterns. *Proceedings of the eighth conference on fire and forest meteorology*, Soc. Am. For., 287-304, Bethesda, Maryland.

[3] Anderson, Hal E. 1982. Aids to determining fuel models for estimating fire behavior. General Technical Report INT-122. Ogden, UT: U.S. Department of Agriculture, Forest Service, Intermountain Forest and Range Experiment Station; 22 p.

[4] Anderson, Hal E. 1983. Predicting wind-driven wild land fire size and shape. Research Paper INT-305. Ogden, UT: U.S. Department of Agriculture, Forest Service, Intermountain Forest and Range Experiment Station; 26 p.

[5] Andrews, P.L. 1986. BEHAVE: fire behavior prediction and fuel modeling system-BURN subsystem, Part 1. Gen. Tech. Rep. INT-194. Ogden, UT: U.S. Department of Agriculture, Forest Service, Intermountain Research Station. 130 p.

[6] Andrews, P.L., C.D. Bevins, R.C. Seli. 2005. BehavePlus fire modeling system, version 3.0: User's Guide Gen. Tech. Rep. RMRS-GTR-106WWW Revised. Ogden, UT: Department of Agriculture, Forest Service, Rocky Mountain Research Station. 132p.

[7] Barros. F.J. 1997. Modeling formalisms for dynamic structure systems. *ACM Transactions on Modeling and Computer Simulation*, 7(4): 501-515.

[8] Barros, F.J. and M.T. Mendes. 1997. Forest fire modelling and simulation in the DELTA environment. *Simulation Probability Theory* 5(3): 185-197.

[9] Birge, J.R. and F. Louveaux. 1997. *Introduction to Stochastic Programming*, Springer Verlag, New York.

[10] Byram, G.M. 1959. Combustion of forest fuels. *Forest Fire: Control and Use*, edited by K.P. Davis, McGraw-Hill, New York, 113-126.

[11] Drysdale, D. 1985. *An Introduction to Fire Dynamics.* John Wiley and Sons. New York.

[12] Finney, Mark A. 1998. FARSITE: Fire area simulator - Model development and evaluation. Research Paper RMRS-RP-4. Ogden, UT: U.S. Department of Agriculture, Forest Service, Rocky Mountain Research Station.

[13] Flood, M. 2001. Laser altimeter: From science to commercial lidar mapping. *Photogrammetric Engineering and Remote Sensing* 67(11): 1209-1217.

[14] Fried, J.S. and B.D. Fried. 1996. Simulating wildfire containment with realistic tactics. *Forest Science* 42(3):267–281.

[15] Gorte, J. and R. Gorte. 1979. Application of economic techniques to fire management - a status review and evaluation. General Technical Report INT-53, USDA Forest Service.

[16] Hu, X. and B. P. Zeigler. 2004. A high performance simulation engine for large-scale cellular DEVS models. *High Performance Computing Symposium (HPC'04), Advanced Simulation Technologies Conference*, April 2004.

[17] Hu, X., A. Muzy and L. Ntaimo. 2005. A hybrid agent-cellular space modeling approach for fire spread and suppression simulation, *Proceedings of 2005 Winter Simulation Conference*, December 3-6, 2005, 248-255.

[18] Hu, X., B. P. Zeigler and S. Mittal. 2005. Variable Structure in DEVS Component-Based Modeling and Simulation. *SIMULATION: Transactions of the Society for Modeling and Simulation International* (81)(2): 91-102.

[19] Hu. X. and L. Ntaimo. 2006. Dynamic multi-resolution cellular space modeling for forest fire simulation, *Proceedings of 2006 Spring Simulation Multi-Conference*, Huntsville, AL, April 2-5, 95-102.

[20] Kim, D., S.J. Buckley and B.P. Zeigler. 1999. Distributed supply chain simulation in a DEVS/CORBA execution environment. In *Proceedings of the Winter Simulation Conference*, Phoenix, AZ.

[21] Miller, C. and D.L. Urban. 1999. Forest pattern, fire, and climatic change in the Sierra Nevada. *Ecosystems.* 2:76-87.

[22] Mittal, S., E. Mak and J.J. Nutaro. 2006. DEVS-based dynamic model reconfiguration and simulation control in the enhanced DoDAF design process. *Journal of Defense Modeling and Simulation (JDMS)*, to appear.

[23] Morais, M. 2001. *Comparing spatially explicit models of fire spread through chaparral fuels: A new model based upon the Rothermel fire spread equation*. MA Thesis, University of California, Santa Barbara.

[24] Mutlu, M., S.C. Popescu, C. Stripling, and T. Spencer. 2006. Assessing surface fuel models using LiDAR and multispectral data fusion. *Remote Sensing of Environment*, In review.

[25] Muzy, A., G. Wainer, E. Innocenti, A. and Aiello, J.F. Santucci. 2002. Comparing simulation methods for fire spreading across a fuel bed. *Proceedings of AIS'2002*, Lisbon, Portugal.

[26] Muzy, A., E. Innocent, A. Aiello, J-F. Santucci and G. Wainer. 2005. Specification of Discrete Event Models for Fire Spreading. *SIMULATION: Transactions of the Society for Modeling and Simulation International* 81 (2): 103-117.

[27] Ntaimo, L., B. P. Zeigler, M. J. Vasconcelos and B. Khargharia. 2004. Forest fire spread and suppression in DEVS, *SIMULATION: Transactions of the Society for Modeling and Simulation International*, 80 (10): 479-500.

[28] Ntaimo, L., and B. P. Zeigler. 2004. Expression of a forest cell model in parallel DEVS and Timed Cell-DEVS formalisms, *Proceedings of the 2004 Summer Computer Simulation Conference*, San Jose, CA, July 25-29.

[29] Ntaimo, L., and B. P. Zeigler. 2005. Integrating fire suppression into a DEVS cellular forest fire spread model, *Proceedings of the 2005 Spring Simulation Multi-Conference*, San Diego, CA, USA, April 3-7: 48-54.

[30] Ntaimo, L. and B. Khargharia. 2006. Two-dimensional fire spread decomposition in cellular DEVS models, *Proceedings of 2006 Spring Simulation Multi-Conference*, Huntsville, AL, April 2-5, 103-109.

[31] Ntaimo, L., W-J. Lee and A. Jalora. 2006. A stochastic mixed-integer programming approach for wildfire containment. *Proceedings of IIE Annual Conference*, Orlando, FL, May 21-24.

[32] Pyne, S.J., P.L. Andrews and R.D. Laven. 1996. *Introduction to Wildland Fire*. 2nd ed. John Wiley & Sons, New York.

[33] Richards, G.D. 1990. An elliptical growth model of forest fire fronts. *International Journal for Numerical Methods in Engineering* 30:1163-79.

[34] Rothermel, Richard C. 1972. A mathematical model for predicting fire spread in wildland fuels. Research Paper INT-115. Ogden, UT: U.S. Department of Agriculture, Forest Service, Intermountain Forest and Range Experiment Station. 40 p.

[35] Rothermel, R.C. and G.C. Rinehard. 1983. Field procedures for verification and adjustment of fire behavior predictions. USDA forest service General Technical Report INT-142. Intermountain Forest and Range Experiment Station, Ogden, UT.

[36] Sun, Y., X. Hu. 2006. Performance Analysis for DEVS Dynamic Structure on Forest Fire Spread Simulation. *Proceedings of the 14th AI, Simulation, and Planning (AIS) conference.*

[37] Uhrmacher, A.M. 2001. Dynamic Structures in Modeling and Simulation - A Reflective Approach. *ACM Transactions on Modeling and Simulation*, Vol.11. No.2 , p. 206-232, April 2001.

[38] USDA Testimony to Congress. Accessed from www.fs.fed.us/congress/108/house/oversight/rey/050504.html, Nov 2006.

[39] Vasconcelos, J. M. 1993. *Modeling Spatial Dynamic Ecological Processes with DEVS-Scheme and Geographical Information Systems*, Ph.D. Dissertation, Dept. of Renewable and Natural Resources, University of Arizona, Tucson, U.S.A.

[40] Wagner, W., A. Ullrich, T. Melzer, C. Briese, and K. Kraus. 2004. From single-pulse to full-waveform airborne laser scanners: potential and practical challenges. *International Archives of Photogrammetry and Remote Sensing*, Vol. XXXV, Instanbul, Turkey.

[41] Wainer, G. and N. Giambiasi .2001. Timed Cell-DEVS: modeling and simulation of cell spaces, In *Discrete Event Modeling & Simulation: Enabling Future Technologies*, Springer-Verlag: 187-213.

[42] Weber, RO. 1991. Modeling fire spread through fuel beds. *Progress Energy Combustion Science* 17:67-82.

[43] Wildland Fire Statistics, National Interagency Fire Center. Accessed from www.nifc.gov/stats/wildlandfirestats.html, Nov 2006.

[44] Zeigler, B. P. 1976. *Theory of modeling and simulation*, Wiley Interscience.

[45] Zeigler, B.P., Y. Moon, D. Kim, and J.G. Kim. 1996. DEVSC++: A high performance modeling and simulation environment. *Proceedings of the 29th Annual Hawaii International Conference on System Sciences*, 1996.

[46] Zeigler, B.P., D. Kim and S.J. Buckley. 1999. Distributed supply chain simulation in a DEVS/CORBA execution environment. *Proceedings of the 1999 Winter Simulation Conference,* 1333-1340.

[47] Zeigler B. P., H. Praehofer, and T.G. Kim. 2000. *Theory of modeling and simulation*, 2nd Edition, Academic Press.

[48] Zeigler B. P. and H. Sarjoughian. 2002. *Introduction to DEVS Modeling and Simulation with JAVA: A Simplified Approach to HLA-Compliant Distributed Simulations*, The University of Arizona, Tucson, Arizona, USA, http://www.acims.arizona.edu.

[49] Zeigler, B.P. 2003. Discrete event abstraction: An emerging paradigm for modeling complex adaptive systems. In *Advances in adaptive complex systems*, edited by L. Booker. Oxford, UK: Sante Fe Institute/Oxford Press.

***Lewis Ntaimo** is an assistant professor of Industrial and Systems Engineering at Texas A&M University, College Station, Texas.*

***Xiaolin Hu** is an assistant professor of Computer Science at Georgia Sate University, Atlanta, Georgia.*

***Yi Sun** is a Ph.D. student in Computer Science at Georgia Sate University, Atlanta, Georgia.*