Burn: A Simulation of Forest Fire Propagation*

Marshall S. Veach† Paul Coddington‡ Geoffrey C. Fox§

Paper for a 1994 NPAC REU Project

Abstract

A project is described which designed, implemented and evaluated a simulation of forest fire propagation as a cellular automaton in a parallel environment. Fire models developed by Richard Rothermel were used as the basis for propagation behavior. Furthermore, the simulation was developed to support a variety of applications including fire propagation prediction and the evaluation of fire fighting effectiveness.

1 Artificial Life, Cellular Automata & Ecological Modeling: An Introduction

1.1 Artificial Life

Artificial life is the study of artificial systems that exhibit behavior characteristic of natural living systems [4]. It is a discipline which attempts to understand the laws that govern natural processes. Although the field has existed, in one form or another, for centuries, it has only recently, with the first artificial life conference in September of 1987, begun to flourish as a discipline. One of the primary reasons for such a slow start has been the need for intensive computing power to model even the simplest of living systems [4]. However, as parallel processing continues to make significant contributions to available computing power, biological simulations have become increasingly viable. As a result, a variety of methods have been developed as tools for modeling life. These include Lindenmayer Systems, genetic algorithms, and cellular automata [4]. In general, these methods emphasize local interactions as opposed to global control. The life–like behaviors these models generate is usually an unprogrammed emergent behavior [4]. And although each of the methods listed above have found application in artificial life, one of them stands out as its premier tool: the cellular automaton [1].

1.2 Cellular Automata

By definition, a cellular automaton is an array of identically programmed automata, or "cells", which interact with one another according to a finite set of prescribed rules [2].

*Project proposed as part of the 1994 Research Experiences for Undergraduates (REU) Program in High-Performance Computing conducted by the Northeast Parallel Architectures Center (NPAC) at Syracuse University. The National Science Foundation provides primary funding for the 1994 NPAC REU Program, through NSF Grant CDA-9200577. Additional funding is provided by the Vice President for Research and Computing, the College of Engineering and Computer Science, the College of Arts and Sciences, the Graduate School, and NPAC, Syracuse University.

†Research Apprentice, 1994 NPAC REU Program; Computer Science Undergraduate Major, Texas A&M University.

‡Research Scientist, Northeast Parallel Architecture Center, Syracuse University.

§Director, NPAC; Professor, Computer Science and Physics, Syracuse University.

In general, a cellular automaton is composed of three primary elements: a state, a neighborhood and a program.

State: A state is a variable associated with each cell in the array. For example, a cell which represents a portion of landscape might have different states which represent the types of animals dominant at that location.

Neighborhood: A neighborhood is the set of cells with which any particular cell interacts. Normally, these are the cells physically closest to the cell in question.

Program: Finally, the program is a finite set of rules which define how a cell's state changes with respect to its current state and the current states of its neighbors.

1.3 Ecological Modeling

Cellular automata, while exceedingly simple to describe, often exhibit surprisingly complex behavior. For example, empirical studies by Wolfram and others have shown that even the simplest of cellular automata can exhibit such emergent traits as self-organization, periodic stability, indefinite growth, irregular contraction, and self-replication [7]. It is this phenomena of emergent life-like behavior which have made cellular automata so popular in artificial life research — both as a metaphor and a tool for implementation. However, all problems are not equal — some provide a problem space better suited for a cellular automata approach than others. For example, the problem of ecological modeling has the potential to take full advantage of the strengths of cellular automata. Consider a cellular automaton model of a landscape consisting of fixed arrays where each cell represents an area of the land surface. The number of cells would fix the scale. And a cell's state could correspond to any number of environmental factors: for example, population, elevation, or terrain. The neighborhood functions could simulate ecological processes such as seed dispersal, epidemic spread or even forest fire propagation. Time could be modeled by each iterative update where the scale depends on the particular process under simulation. Green points out that such an approach has many advantages. The cellular nature of the model makes the approach compatible with pixel–based satellite imagery or with quadrat–based field observations. Furthermore, it allows processes that involve movement through space and time to be modeled intuitively and easily. In fact, cellular automata models have been successfully implemented in a variety of ecological modeling applications including starfish outbreaks, spread of disease, and forest dynamics [1].

2 Fire Propagation Simulation: Background & Description

While several computer simulations of fire propagation currently exist, none provide a comprehensive model designed for use with multiple terrains. Some are designed for use with general terrains but lack an acccurate fire model. Others have a sophisticated simulation that succeeds only with specific predetermined terrains. However, many of the models do have a common element. Many base their simulations, to some degree, on a set of fire spread models designed by Richard Rothermel in 1983 [6].

What follows is a general description of a simple fire spread model which is based on Rothermel's equations. Although I have not included the complete set of formulas, all necessary calculations can be done via direct computation or nomograms. In general, a fire prediction process has three elementary components: obtaining input, calculating fire behavior, and interpreting output. However, due to the large number of relatively

uninteresting, detailed formulas required to relate the calculation procedure, I will, in this paper, merely describe the most important of those equations.

Fire Model Input

Many factors influence the behavior of a forest fire. However, some factors are more influential than others. There are four primary behavior inputs: fuel type, fuel moisture, wind and slope.

1. **Fuel types** determine how different terrains affect fire spread. A fuel type can be described using the following parameters:
   - Fuel Loading - the mass of fuel per unit area, live and dead, grouped by particle size classes.
   - Fuel Depth - ft
   - Fuel Particle Density - $\frac{lb}{ft^2}$
   - Heat Content of Fuel - $\frac{Btu}{lb}$
   - Moisture of Extinction - the upper limit of fuel moisture beyond which the fire will no longer spread uniformly.

   While, in theory, a fuel type could have associated with it any value for the above parameters, most terrains can be described by one of thirty-three different parameters combinations. These parameter combinations are referred to as the primary fuel types.
2. **Fuel moisture** has a tremendous impact on the intensity of a fire and the rate of its spread. A fire model must account for the fuel moistures' affect on both fire intensity and the heat necessary to bring the fuel ahead of a spreading fire up to ignition temperature. In addition, one must consider the separate fuel moistures for both live and dead fuels. The best source of fuel moisture data is a current record from a National Fire Danger Rating station. However, methods do exist for relating environmental factors to fuel moisture. These factors include: relative humidity, intensity of recent rainfall and the types and developmental stages of various fuel types.
3. **Wind** provides the most variable influence on fire behavior. The unpredictable nature of wind derives from the large number of distinct classes of wind and from its ability to change both horizontal and vertical direction. The necessary wind data required for predicting fire spread include the type and velocity of wind at the height above the surface equivalent to half the height of the fire at all points where differing fuel types, slopes or moisture imply unique fire behavior.
4. **Slope**, also, has an important role in fire spread. For example, a fire spreads much more quickly up a slope than it does on a level terrain. In addition, the length of the flame will be significantly higher on a slope. As such, an accurate fire model must account for the change in elevation over a modeled terrain. Fortunately, calculating slope from topographic maps is a straightforward procedure.

Fire Model Output

Given the appropriate inputs, a fire model should yield a variety of outputs that can be used to predict the behavior of the fire. These outputs include the following (equations have been included for the more important outputs [5]):

- **Rate of Spread**, $\frac{Ft}{Min}$

$$R = \frac{(IR)(Flux)(1 + WindCoefficient + SlopeCoefficient)}{(BulkDensity)(HeatingNumber)(PreignitionTemp)} \quad (1)$$

The rate of advance of the head of a fire.

- **Preigntion Tempature**, $\frac{Btu}{lb}$

$$PreignitionTemp = 250 + 1,116 MoistureContent \quad (2)$$

The tempature to which a fuel must be heated in order for ignition to occur.

- **Heat per unit area**, $\frac{Btu}{ft^2}$
The amount of heat released per square foot during the time that area is within the flaming front.
- **Fireline intensity**, $\frac{Btu}{ft/s}$
The amount of heat released per foot of fire front per second - this has a direct relationship on the difficulty to contain a fire.
- **Flame length**, ft
Average length of flame at a projection point - can be used as measure of fireline intensity.
- **Spread distance**, ch
Estimation of probable forward movement of the head of a fire during a specified time period.
- **Map distance**, in
This is an estimate of the progress of the fire front.
- **Perimeter**, ch
Estimation of perimeter of fire.
- **Area**, a
Estimation of area of fire.
- **Reaction intensity**, $\frac{Btu}{ft^2/min}$

$$IR = (OptRctnVel)(NetFuelLoad)(FuelPartLowHeat)(MoistDamping)(MinDamping) \quad (3)$$

This is the rate of heat released per square foot per minute.

In terms of implementation, the inputs described in this section yield the outputs which will be, in turn, used to control a cellular automata. Any given cell which is on fire has a certain probability of igniting adjacent cells or sending a firebrand downwind. This probability varies as a function of the outputs.

3 The Implementation

At the time of this writing, an Alpha version of Burn is implemented in the parallel language of Fortran90. Burn models fire spread using Rothermel's equations described earlier in this paper. There are however some difference between Rothermel's model and the model that Burn incarnates. In order to understand these differences, consider, first, the particulars of Burn's implementation. Such a discussion is best divided into several distinct sections:

The Burn Space

Burn space refers to the area in which a burn occurs. For example, the burn space of a forest fire is the area in which the fire can spread. This space is usually bounded by some sort of natural boundary such as an ocean, river, mountain or chasm. For the purposes of simulation, however, the burn space is bounded by a more artificial barrier: memory. In this particular simulation, the burn space consists of a 100x100 matrix. Each cell in the matrix represents a square area of land 20 feet long and 20 feet wide. Thus, each cell represents 400 square feet of land. And the entire burn space consists of 10,000 of these cells creating a total area of 40,000 square feet. While this burn space size may seem limiting, its size is entirely arbitrary. In fact, a two-dimensional matrix of any configuration would accomodate the model just as well as the test space.

Burn Input

Burn takes its input in the form of ASCII text files. Currently, the program expects to find five input files: a terrain file, a moisture data file, a wind data file, a elevation data file, and an ignition data file. The terrain file contains a value for every cell of the burn space. Burn uses this value to determine which terrain parameters to associate with that particular cell. The moisture data contains a value for each cell which represents the percent moisture for the fuel in that particular cell. Similarily, the wind and the elevation data have values for each cell. The first step of execution in Burn involves reading these values from ASCII files.

Burn Calculations

As mentioned previously, Burn's fire spread model is based on a set of equations developed by Richard Rothermel. Unfortunately, those equations are intended for use with homogeneous fuel beds. In order to model fire spread in a heterogenous fuel bed, Burn employs a unique method of "jumping". The fire models spread within a cell using Rothermel's original equations. For every cell, the fire always ignites in the exact center of the cell and spreads towards the edges of the cell. However, when the fire reaches a boundary between two distinct fuel types, Rothermel's equations provide no way to model the transistion. In order to compensate for this deficiency, Burn assumes that the fire jumps from the edge of cell about to ignite into the center of that cell. While this method has no parallel in actual fires, it provides a simple and effective procedure for modelling heterogeneous fuel beds.

4 Application of Burn

To demonstrate the potential applications of Burn, the author used the Burn platform to perform two experiments. These experiments were designed to accomplish a number of different objectives. First, they should provide some insight into the accuracy of Burn's fire model. Second, they should provide some indication of how easily Burn can be used to provide insight into fire spread. Third, they should answer some questions about the nature of fire spread. The two experiments that will be detailed in this paper, accomplish all three of these objectives. The third part of this section describes possible future experiments other than the ones actually performed for the completion of this paper.

Experiment One: Moisture Content

It is well known that the amount of moisture present in a fuel has an affect on its ability to burn. By increasing the ignition tempature (the tempature at which a fuel ignites)

of a fuel, water slows down the fire's rate of spread. And, in fact, every fuel has a moisture extinction threshold. This threshold is the percent of water absorption that will yield the fuel entirely non-flammable. While all of these conditions have been demonstrated clearly, the exact relationship between fire spread and fuel moisture content remains unknown. Fortunately, Burn provides an ideal platform to explore this relationship. By being able to model many different fires in a relatively short amount of time, Burn provides an excellent platform for comparing the relative effects of varying input configurations. In this case, the experimenter was interested in evaluating the significance of the effect that moisture content of a fuel has on rate of spread. In order to test this attribute, all parameters, except for moisture content, were held constant. In this case, the fuel bed was a completely random distribution of every fuel type. This created a tremendously heterogenous fuel bed. Furthermore, in order to further isolate the effects of moisture, the slope and the wind were both assumed at zero. Also, the fire was ignited, for every burn, in the precise center of the burn space. Finally, the moisture extinction content was set at a constant .30. These inputs were then coupled with universal moisture contents from 0 to .29 by increments of .01. As expected, the time, in minutes, neccessary to fill the burn space increased with the moisture content of the fuel. However, what seems like a tremendously linear increase in time at low levels of moisture content becomes distinctly exponential as the moisture contents approach the extinction level. This exponential increase occurs due to the fact that at extinction level, the fire will not burn which makes the time neccessary to fill the burn space infinite.

Experiment Two: Burn Space Topology

While significant amounts of research have been devoted to the improvement of the mathematical models for fire spread, very little effort has been spent on more global analyses of fire spread. For example, most fire spread models compute rate of spread independent of the fire as a whole. The models yield results on an entirely local level. This makes the models ideal for implementation via cellular automaton, but it denies the reality of the situation: attributes such as rate of spread are correleated to properties of the fire as a whole. This should be obvious enough. A fire's rate of spread is dependent on the area of ignitable fuel exposed to the perimeter of the fire. Thus, the topology of the fire provides significant information about the fire's attributes. Moreover, the topology of the burn space provides information about the possible topology of the fire. Unfortunately, the local nature of previous fire model research has resulted in no serious effort investigating the impact of burn space topology on fire spread. As mentioned earlier, all burn spaces are bounded (eventually) by some barrier, usually natural. In the Burn simulation, these barriers are the bounds of the burn space matrix. The question is: How do these barriers affect fire spread? In order to answer this question, the author ran several fire simulations with the only unique variable being the location of the ignition point within the burn space. In this case not only were the fuel types distributed randomly but the moisture content of each cell was determined randomly. As expected since the burn space was square, the time neccessary for burn completion decreased as the location of the ignition point moved towards the center. Moreover, ignition points with identical distances from the center had approximately identical burn times despite the differences in terrain and fuel types for those ignition points.

Future Experimentation

The two experiments described in this section were designed primarily to test Burn's application potential. Clearly, Burn can be used in such a manner. In fact, the potential for future experimentation is virtually infinite. One of the primary uses for Burn will be to model multiple variable inputs simultaneously. For example, consider the effects of moisture and ignition point location simultaneously. Does one input tend to neutralize the other? Do they seem to have independent effects? Or perhaps they interact to mutually enhance their individual effects. In addition to these questions, Burn can be used to determine the relative danger ratings of various input configurations. How dry is too dry? Where is the worst place for a fire to start? What is the worst topology for a burn space? Finally, Burn can be used to model fire fighting techniques. By changing the terrain dynamically, Burn can simulate the placing of fire breaks and the placement of fire fighting units. In doing so, Burn can answer questions like: Where is the best location for a unit or break? Where is the fire going to be most out of control? Where will the fire not need a unit or a break? Hopefully, by providing answers to questions like these Burn can contribute significantly to our understanding of how fire spreads.

5 Conclusion

The principles of artificial life and cellular automata can be applied to the problem of fire spread modeling. The successful implementation of Burn has established this fact. Moreover, such a model offers unique advantages over previous models. First, it is efficient; completing its work in tractable time. Second, it provides a unqiue perspective to fire modeling. Specifically, it provides a global perspective from a local model. And while further , more detailed, experimentation will be neccessary before any final conclusions can be drawn, the prelimenary results described in this paper are very encouraging. It seems to bridge the gap between local control and global behavior. Third, it brings simplicity of design to a complicated model. That is. it makes it easy to identify the important parameters of fire spread without having to study the details of computation. Clearly, Burn's current success promise further results. Hopefully, as more complicated and detailed experimentation is performed using the model, its sophistication will continue to grow.

6 Acknowledgements

Thanks to my advisors Dr. Paul Coddington and Dr. Geoffrey Fox. Also thanks to Dr. Ed Bogucz, who seems to serve as a virtual advisor to all. In addition, thanks are in order for the entire NPAC faculty and staff. And finally, thanks to the National Science Foundation, whose funding made this summer possible.

References

[1] Green, D. G., "Connectivity and complexity in landscapes and ecosystems", *Pacific Conservation Biology*, in press, 1994.

[2] Green, D. G., "Cellular Automata", preprint, 1993.

[3] Hargrove, W., *Simulating Fire Patterns in Heterogeneous Landscapes*, Unpublished manuscript available via World-Wide-Wed, URL=//http://jupiter .esd.ornl.gov/ern/embyr/embyr.html

[4] Levy, S., *Artificial Life*, Vintage Books, 1992.

[5] Rothermel, R., " A mathematical model for predicting fire spread in wildland fuels." Res. Pap. INT-115. Ogden, UT: U.S. Department of Agriculture, Forest Service, Intermountain Forest and Range Experiment Station; 1972.

[6] Rothermel, R., "How to predict the spread and intensity of forest and range fires." Gen. Tech. Rep. INT-143. Ogden, UT: U.S. Department of Agriculture, Forest Service, Intermountain Forest and Range Experiment Station; 1983.

[7] Wolfram, S., "Cellular Automata as models of complexity", *Nature*, Volume 311: 419-424, 1984.