
Ring of Fire

Simulating Wildfire Spread Using
Evolved Cellular Automata



Isaac Caruso, Oliver Baldwin Edwards, and Martin Glusker

Why Evolution?



- Massive wildfires over the past year in Australia, California, the Amazon
- All existing simulations rely on same equations predicting fire spread
- What if these are missing something big?



Why Cellular Automata?

- Traditional wildfire simulation is computationally expensive
- Cell2Fire uses cellular automata to simulate wildfire
- We sought to do better with evolution (“Rules for Fire”)

Cell2Fire: A Cell Based Forest Fire Growth Model

Cristobal Pais^a, Jaime Carrasco^b,

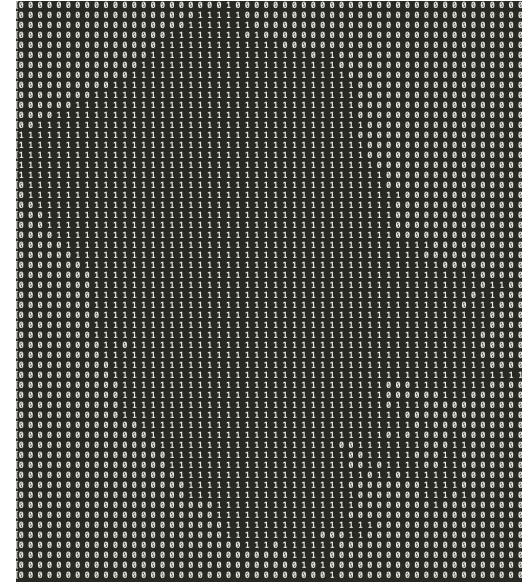
David L. Martell^c, Andres Weintraub^b, David L. Woodruff^d

May 24, 2019

The Data

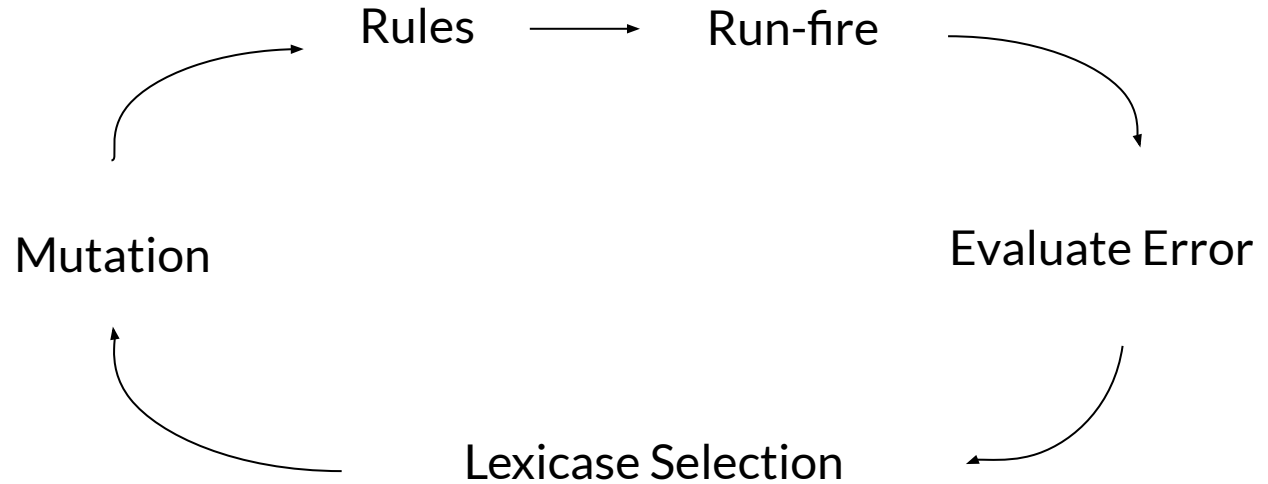
- 10 fires with the following data:
 - Weather (FWI, BUI, ISI)
 - Terrain (Ignition Point, Slope, Forest Composition)
 - Final “Fire Scar”
- 7 fires for train set, 3 for test set
- New variables:
 - Net time neighbors burning
 - Net burning direction
- All used as instructions for our Rules

Sample Fire Scar



Mica Creek Fire 1

Methodology



Rules

```
(def fire-instructions
  (list
    'slope
    'ISI ;; Initial Spread Index
    'BUI ;; Buildup Index
    'FWI ;; Fire Weather Index
    'NT  ;; Neighbor Average Time Burning
    'NBD ;; Net Burning Direction
    'WS  ;; Wind Speed
    'WD  ;; Wind Direction
    'TB  ;; Time Burning
    ':split
    ;; and some propel ones...
  ))

(def a-program '(exec_dup (boolean_and WS boolean_not NT WD))
```

Our Implementation

```
(defn fire-error-function
  "Calls run-fire on each fire in test set... and more on next slide ")
  ;; ("kootenay1"      {:k1 [[0 0 1 0]      :a2 [[1 1 0 0]
  ;;  "arrohead2") →    [1 1 1 0]      [0 1 1 1]
  ;;                    [0 1 0 0]      [1 0 1 0]
  ;;                    [0 1 0 0]]      [0 1 1 1] }
  ;;
  ;;
  ;; Fire names → Map of fire scars

(defn run-fire
  "Runs a fire for 24 hours and returns the final fire scar. ")
  ;; [[0 0 0 0]      [[0 0 1 0]
  ;;  [0 1 0 0] →    [1 1 1 0]
  ;;  [0 0 0 0]      [0 1 0 0]
  ;;  [0 0 0 0]]      [0 1 0 0]]
  ;;
  ;;
  ;; Ignition pt. → Fire scar
```

Our Implementation

```
(defn update-grid
  "Updates a fire grid from one time step to the next"
  ;; [[0 0 0 0]      [[0 0 0 0]
  ;;  [0 1 0 0]    →  [0 1 1 0]
  ;;  [0 0 0 0]      [0 0 0 0]
  ;;  [0 0 0 0]]      [0 0 0 0]]
  ;;
  ;; Minute 1    →   Minute 2

(defn update-cell
  "Updates a cell to its next state by interpreting a push program"
  ;;      0      →    0 || 1
  ;;      Unburned →    Unburned or Burning
  ;;
  ;;      1      →    1 || 2
  ;;      Burning  →    Burning or Burned
```

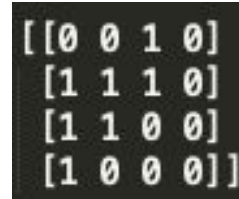
Sample Evaluate Error



A 4x4 matrix of binary values:

$$\begin{bmatrix} [0 & 0 & 1 & 0] \\ [1 & 1 & 1 & 1] \\ [0 & 1 & 0 & 0] \\ [0 & 1 & 0 & 0] \end{bmatrix}$$

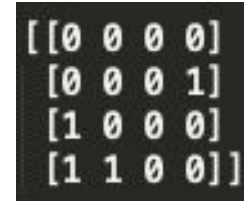
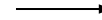
Run-fire output



A 4x4 matrix of binary values:

$$\begin{bmatrix} [0 & 0 & 1 & 0] \\ [1 & 1 & 1 & 0] \\ [1 & 1 & 0 & 0] \\ [1 & 0 & 0 & 0] \end{bmatrix}$$

Fire Scar



A 4x4 matrix of binary values:

$$\begin{bmatrix} [0 & 0 & 0 & 0] \\ [0 & 0 & 0 & 1] \\ [1 & 0 & 0 & 0] \\ [1 & 1 & 0 & 0] \end{bmatrix}$$

Error Vector

Our output: [0010111101000100]

Fire scar: [0010111011001000]

Error Vector: [0000000110001100]

Results

Number of Fires: 4

Population size: 10

Best program:

```
(exec_dup (boolean_and WS boolean_not 2 integer_* integer_ = TB
integer_% NT WD WD) 0 WD boolean_not FWI FWI exec_dup (1 integer_+ TB
false) exec_dup (WS))
```

Best total error: 6877

Total number of cells: 24280

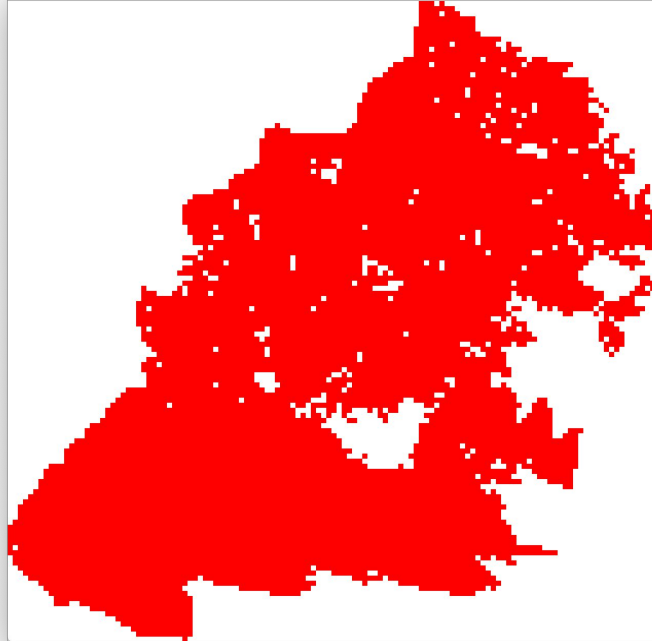
Best percent error: 0.28323722

Training set:

Fires evaluated: (r1 g2 m1 k2)

Arrowhead 1 - Results

35.5% Error



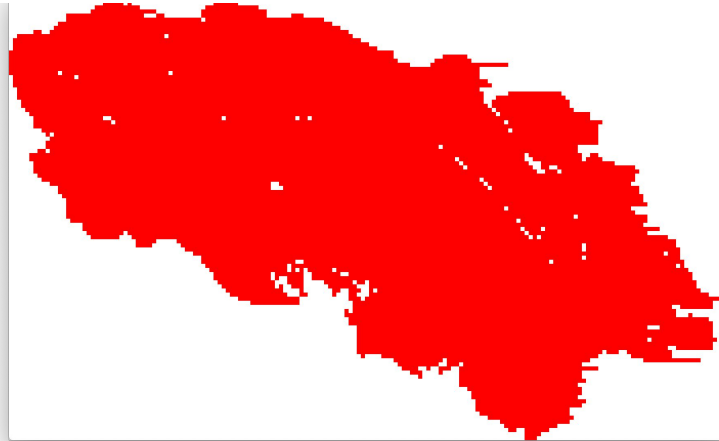
Fire Scar



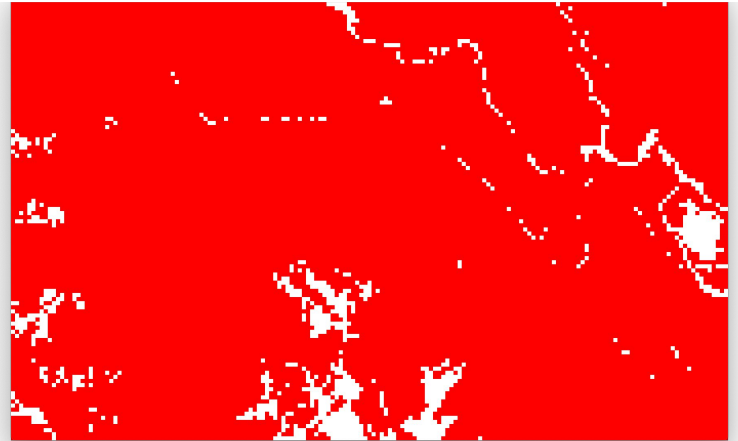
Our Output

Glacier Creek 1 - Results

40.3% Error



Fire Scar



Our Output

Limitations

```
152 -----
153 Report for Generation 15
154 -----
155 Best plushy: (ISI TB false false integer_= NBD boolean_= :split NBD 1 :split boolean_= integer_-)
156 Best program: (ISI TB false false integer_= NBD boolean_= :split NBD 1 :split boolean_= integer_-)
157 Best total error: 3270
158 Total number of cells: 12284
159 Best percent error: 0.26619995
160 Fires evaluated: (k2 m2)
161
162 -----
163 Report for Generation 16
164 -----
165 Best plushy: (ISI TB boolean_and 0 integer_= integer_= exec_if :split integer_= NBD NBD)
166 Best program: (ISI TB boolean_and 0 integer_= integer_= exec_if (:split integer_= NBD NBD) ())
167 Best total error: 3270
168 Total number of cells: 12284
169 Best percent error: 0.26619995
170 Fires evaluated: (m2 k2)
171
172 -----
173 Report for Generation 17
174 -----
175 Best plushy: (ISI TB TB false integer_= :split NBD :split NBD 1 1 :split)
176 Best program: (ISI TB TB false integer_= :split NBD :split NBD 1 1 :split)
177 Best total error: 2631
178 Total number of cells: 7532
179 Best percent error: 0.34930962
180 Fires evaluated: (m1 r1)
181
```

- Not enough generations run
- Complicated to reduce time-complexity with larger programs
- Only final fire-scars, no intermediate time steps
- Only weather data by the hour



Looking Forward

- KEEP EVOLVING!!
- Continue optimizing run time
- Look at implementing ROS metric as instruction
- Continue refining instructions used for each cell



Thank you!

