Applied Geocomputation and Spatial Analysis

Minimum Travel Time based Fuel Treatment Optimization Algorithm in Cellular Automata Forest Fire Simulations

ID: 1427801

Word Count: 2495

April 23, 2017

Minimum Travel Time based Fuel Treatment Optimization Algorithm in Cellular Automata Forest Fire Simulations

1427801

April 23, 2017

1 Introduction

Forest fires are becoming an ever crucial component of natural hazard risk management, as the warming climate introduces unpredictable changes to the world's dynamic and sensitive forests (Bonan et al. 2008, Kurz et al. 2008). Mark Finney and the US Forest Service created a fire paths calculation algorithm to calculate the speed and direction in which simulated fires would spread using ellipsoidal fire networks and generate minimum travel times for fire paths through network searching within lattices (Finney 2002). This algorithm generates accurate fire paths, but as noted by Finney himself, the efficiency of the algorithm for fire suppression has never been tested, which underscores the need for a cellular automata (CA) and shortest path algorithms to test the efficiency of minimum travel time models (Finney 2002, Finney 2006). This study aims to show whether or not Mark Finney's minimum travel time algorithm is a robust metric for the implementation of fuel treatment optimization for passive forest fire suppression through CA modeling (Finney 2002, Finney 2006). Such a CA model could be easily applied to raster DEM maps with proper coding and weighing of cells to simulated forest fire burn paths, burn probabilities, average fire size and most traveled paths to optimize fuel treatments in the real world (Keane et al. 2010, Encinas et al. 2007, Clarke et al. 1994).

2 Literature Review

Climate change has seriously affected the intensity and frequency of forest fires throughout the world. Several researchers have noted that the introduction of abnormally warm climates in the upper latitudes has brought with it an increased voracity in pest insects, low soil moistures and built up flammable debris (Bonan et al. 2008). These new factors in forest fires contribute to the increasing cost of forest management and preventative fire fighting which is now costing North American governments billions of dollars (Headwaters Economics 2013). Models of forest fires like Finney's minimum travel time algorithm and CA forest fire models help to optimize such operations.

2.1 Forest Fires and Climate Change

Climate change is set to dramatically intensify forest fire seasons, especially in North America and South East Asia through lowering soil moisture and increasing the supply of fuel through forest die off (Millar et al. 2007, Kurz et al. 2008, Santilli et al. 2005, Davidson and Janssons 2006). The increase in fuel supply from dead wood created by pest infestations, and the lowered soil moisture often overcomes the climate change induced increased precipitation, and extends and intensifies the fire season (Stocks et al. 1998, Flannigan 2000). Dale et al. (2001) showed in their climate modeling of North America an increase of 30% in seasonal severity ratings over much of Alaska and northern Canada. These problems represent the conditions under which forest fires become an epidemic crisis that threatens the stability of the ecosystems we rely on to help sequester carbon (Dale et al. 2001, Lindner et al. 2010).

2.2 Forest Fire Modeling

Modeling of forest fires is done either in partial differential equations (PDE), or cellular automata (CA) local rules (Karafyllidis and Thanailakis 1997).

PDE's, while more sophisticated, are calculation heavy and end up being inefficient at quickly simulating wildfires (Karafyllidis and Thanailakis 1997). CA models are simplified versions of reality parceled out into matrices of cells with rules and weights determining the interactions between the cells in order to simulate a scenario through time (Wolfram 1983, Wolfram 1984, Finney 2002). Therefore, CA models are popular amongst forest fire researchers for their speed and extensible complexity. Karafyllidis and Thanailakis (1997) created a comprehensive local rule that takes into account the height, fuel and wind influences in directing the growth of forest fires in a CA model. Finney (2002, 2004) created a minimum travel time algorithm to detect the shortest possible path fires take after a simulated wildfire is over. Ager et al. (2007) took this concept and applied it to wildfire risk in Ohio, USA, and was able to extract the necessary regions of fuel clean-up that would help reduce the risk of wildfires (Encinas et al. 2007). This study aims to apply similar methodology and test Finney's algorithm for fuel optimization in suppressing simulated forest fires.

3 Methodology

This study uses a CA model, shortest path algorithms and linear regression analysis to explore the relationships between fuel treatments, burn travel times and burn probabilities in simulated forest environments. Using CA models accomplishes modeling forest fires with relative speed. Although the resultant models created aren't necessarily the most accurate, if run through hundreds of iterations, they provide an adequate modeled scenario to begin formulating larger, more complex models to analyze forest fires with greater accuracy (Wolfram 1983, Wolfram 1984, Finney 2004). Furthermore, the methods used in analyzing burn patterns across relatively small but descriptive CA parameter matrices can later be applied to raster elevation maps easily, as the CA models being used already incorporate elevation into their structure.

3.1 Karafyllidis and Thanailakis (1997) Local Rule

The CA model created uses the base logic Karafyllidis and Thanailakis (1997) outlined in their CA model on wild fire propagation. It takes into account the interactions between cells with equal side's of length a and incorporates diagonal cell interaction and wind influence into the fire propagation. In addition, this study's model uses randomly generated elevation value and fuel type value matrices to influence the fire propagation. The CA model uses normal spatially random fire ignition to start the forest fire simulation. The fire propagation uses the local rule as described below:

$$\begin{aligned} S_{i,j}^{t+1} = R_{i,j}(S_{i,j}^{t} + (&nH_{i-1,j}S_{i-1,j}^{t} + wH_{i,j-1}S_{i,j-1}^{t} \\ &+ eH_{i+1,j}S_{i+1,j}^{t} + sH_{i,j+1}S_{i,j+1}^{t}) \\ &+ 0.83(nwH_{i-1,j-1}S_{i-1,j-1}^{t} + neH_{i-1,j+1}S_{i-1,j+1}^{t} \\ &+ swH_{i+1,j+1}S_{i+1,j+1}^{t} + seH_{i+1,j-1}S_{i+1,j-1}^{t})) \end{aligned}$$

Where S is the state; t, the iteration or time step; i and j, the locations of the cells; R, the spread rate; H, the height difference of the local cell; n, s, e, w, nw, ne, sw, se, the wind influence for each cardinal direction; 0.83, the constant weight calculated for the local diagonal cells which allows for circular propagation of the forest fires. After initially generating a terrain and fuel landscape, these parameters are held constant throughout the simulation in order to prevent variability in the scenarios, as the aim of this study is mainly the optimization of fuel treatments across the scenarios. Wind velocity, duration and ignition points are normal randomly generated to create variability in fire sizes, and wind direction is held constant at normal random north westerly and north easterly directions. Eleven different fuel types are included to describe a variety of fuel sources, they are multipliers (between 0.01 and 0.1 m/s) to influence the speed at which the fire moves through a certain cell. The scenario set up was a 400×400 cell grid. They are described in Figure 2a.

3.2 Burn Probability Matrix

After $n = 1000$ simulations, the resultant matrices of burnt and unburnt cells are used to generate a burn probability matrix for the specific fuel treatment in which the number of times each cell is burnt (cell becomes 1) is summed and divided by the total number of simulations run and multiplied by 100 for a percentage burn (Finney 2002, Keane et al. 2010):

$$BP_{i,j} = (b/n) \times 100$$

Where BP is the burn percentage; i and j, the locations of the cells; b, times burnt; n, the sample size of simulations run.

3.3 Minimum Travel Time Burn Paths

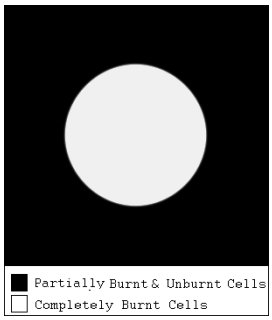
In order to analyze burn paths most frequently used per simulation, a shortest path algorithm is applied to calculate the shortest path for each simulation from the origin (ignition point), to the end of the fire (the edges of the burnt area). To do this, an edge detection algorithm is first be applied to extract an array of the end points of the fire, and the shortest path algorithm is applied. The burnt network of cells is represented as a graph, where the burnt cells (nodes) are connected to adjacent cells through edges (Finney 2002). The edge weight is determined by the fuel influence, elevation and wind direction as described in the below equation:

$$W_{i,j,(i,j-1)} = D_{i,j,(i,j-1)}/(wF_{i,j-1}(H_{i,j} - H_{i,j-1}))$$

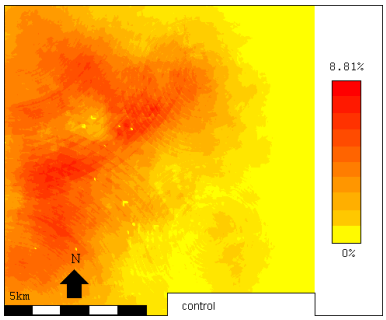
Where $W_{i,j,(i,j-1)}$ is the edge weight from i, j to $i, j-1$, $D_{i,j,(i,j-1)}$ is the horizontal distance between i, j and $i, j-1$, $F_{i,j-1}$ is the fuel influence at node $i, j-1$, H is the height, and w is the wind influence. The shortest path algorithm starts at the origin, searches for the greatest burn value in the surrounding cells, and traces the burn path to the edge (See Appendix 1).

These calculated burn paths are then aggregated and a mean length of burn paths and a heat map of the most traveled paths is created. A

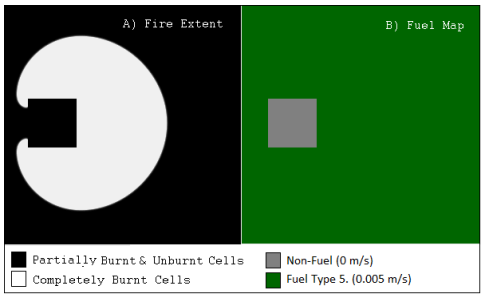
mean burn probability for the cells in the simulation, mean length of burn paths, mean fire size, max fire size, as well as the largest 10% fire sizes and burn probabilities are extracted for analysis.



(a) Local Rule Test

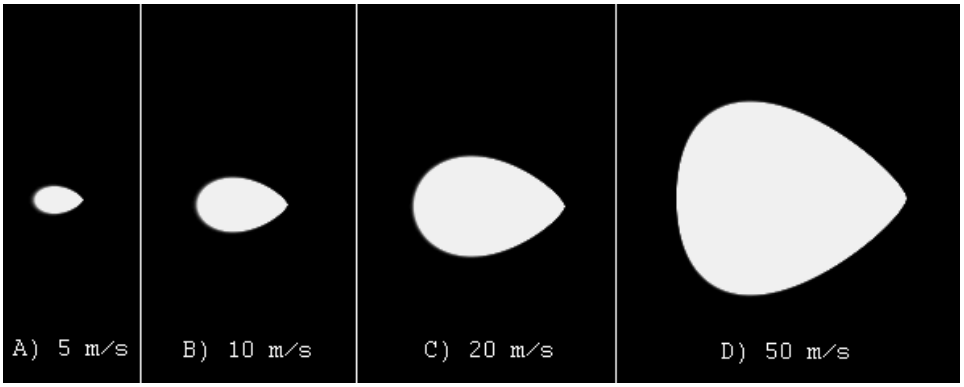


(b) Burn Probabilities Test



(c) Fuel Types Test

(d) Topography Test



(e) Wind Test

Figure 1: CA model components testing

3.4 Fuel Treatment Optimization & Regression Analysis

Each subsequent 1000 simulations of the CA model has fuel treatments applied on the most traveled burn paths and highest burn probability cells. Five treatment scenarios are simulated, where 1% to 5% of the most burned paths and cells are treated, adjusting their fuel types to the "non-fuel" type that do not burn in the simulation. Fire sizes and burn probabilities are then extracted for regression analysis against the fuel treatment intensity.

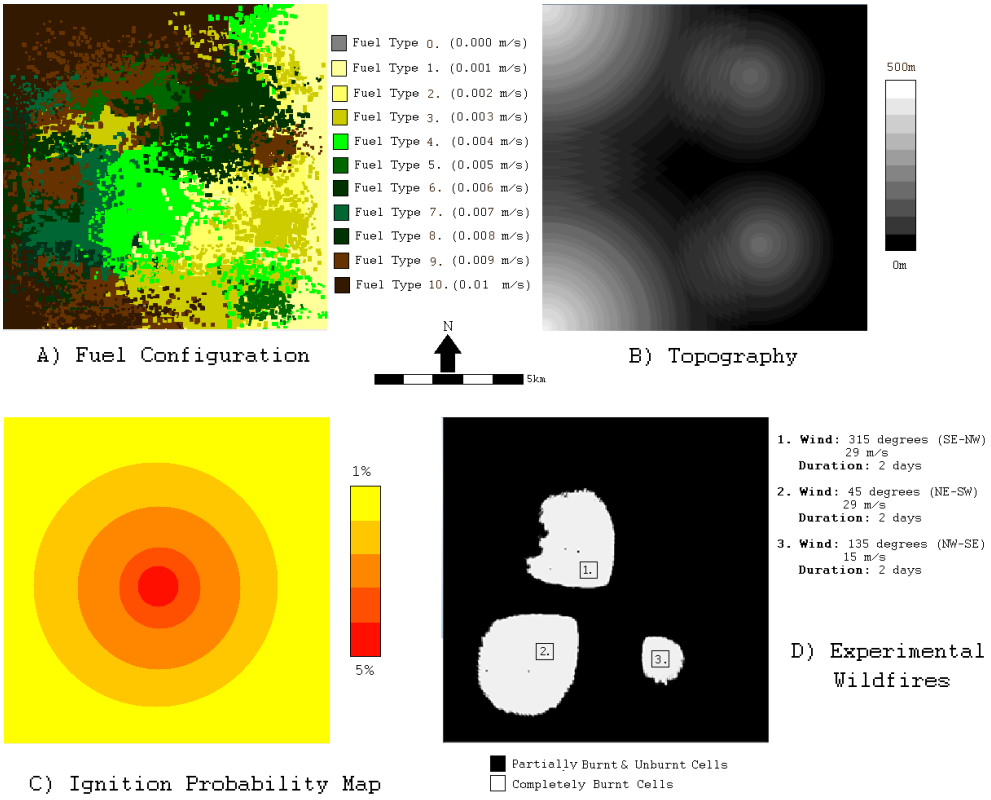


Figure 2: CA model parameter configurations

4 Results

4.1 MTT Use for Fire Suppression in CA Modeling

The CA model with minimum travel time based fire suppression intervention showed the fire sizes, mean burn probabilities of cells, and fire path lengths all fell with increasing percent intervention (Figure 3). This was expected as the suppression, at first glance, seemed to successfully prevent the spread of the fires (Figure 4 and 5).

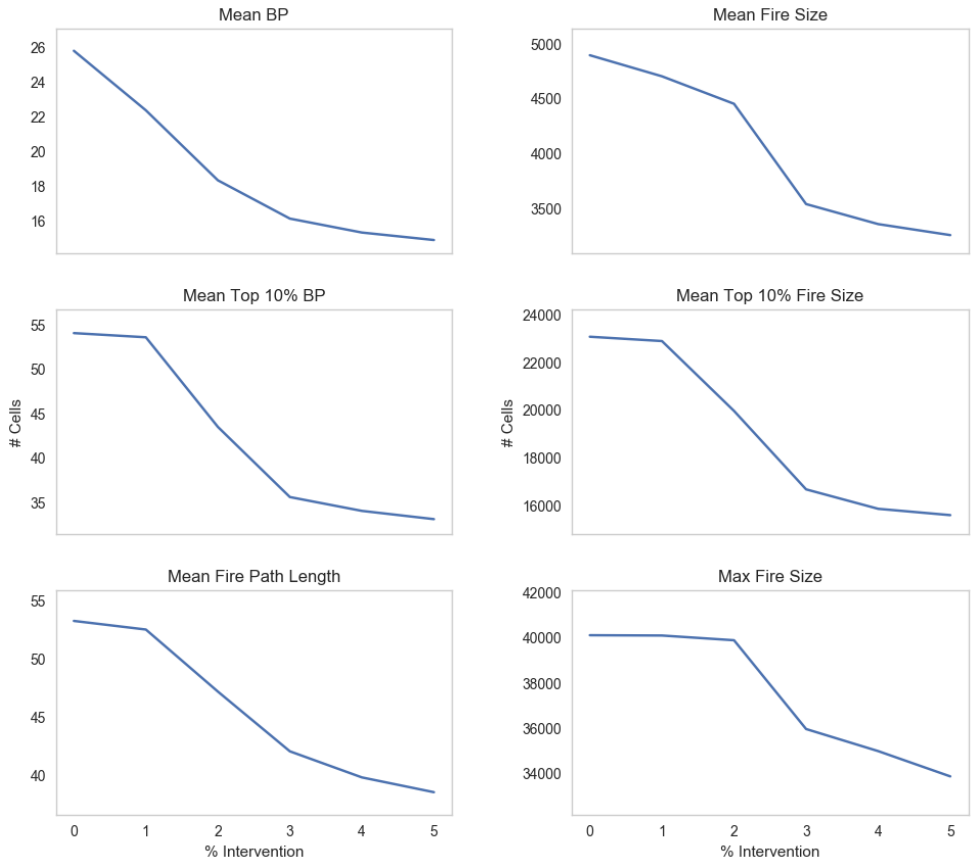


Figure 3: Changes in BP, Path Lengths and Fire Sizes

The fire suppression contributed up to 8% lower burn probabilities along the suppressed fire paths, and saw noticeable changes throughout the simulated scenario outside the targeted burn paths (Figure 5 and 6).

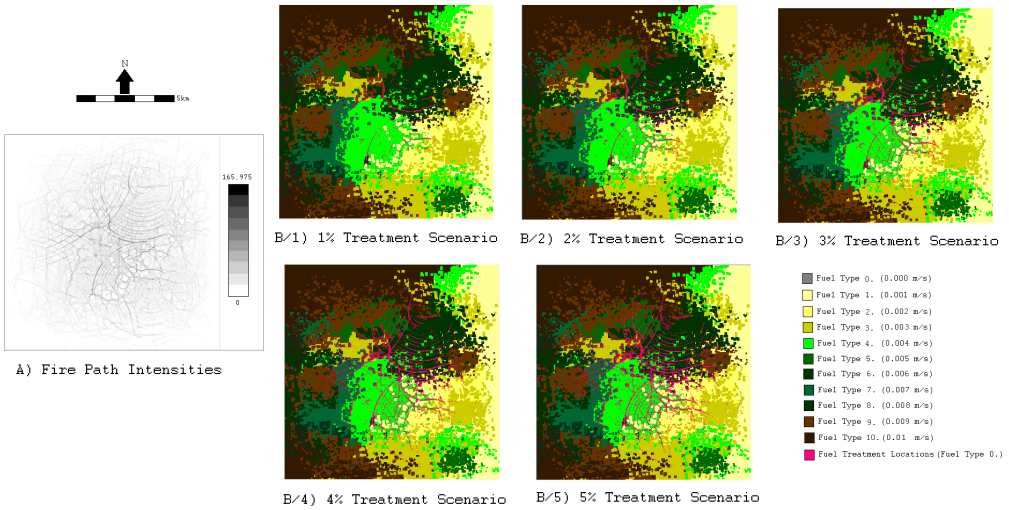


Figure 4: Fire Paths and Fuel Treatment Locations

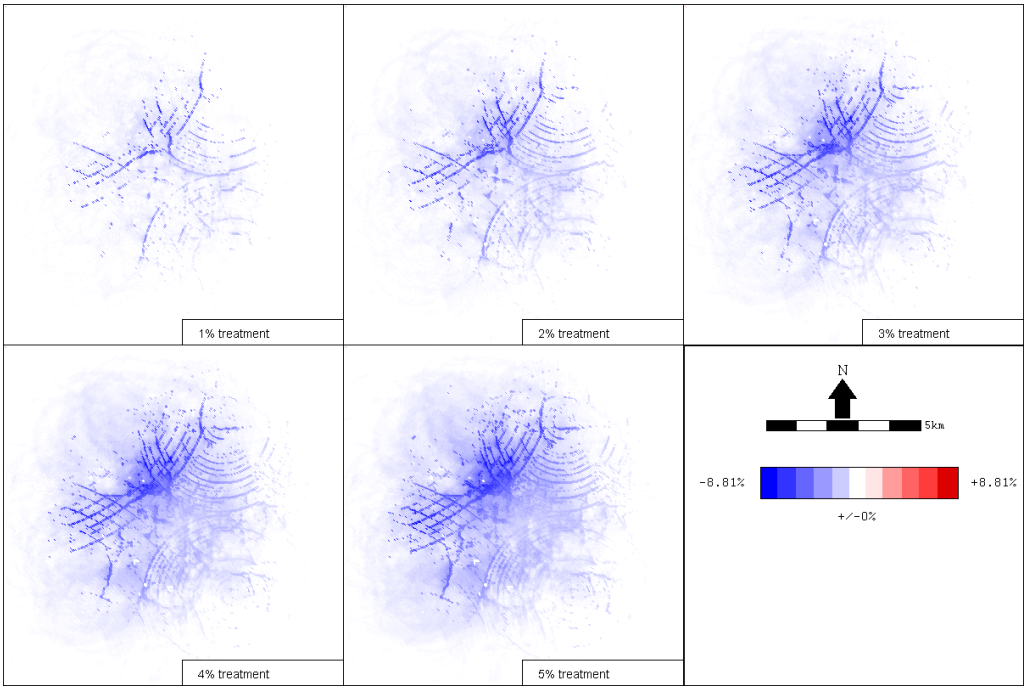


Figure 5: Fuel Optimization Percent Change in BP

4.2 Regression Analysis of CA model Results

The linear regression analysis of the change in fire sizes and burn probabilities (BP) came up inconclusive. While the intervention was statistically

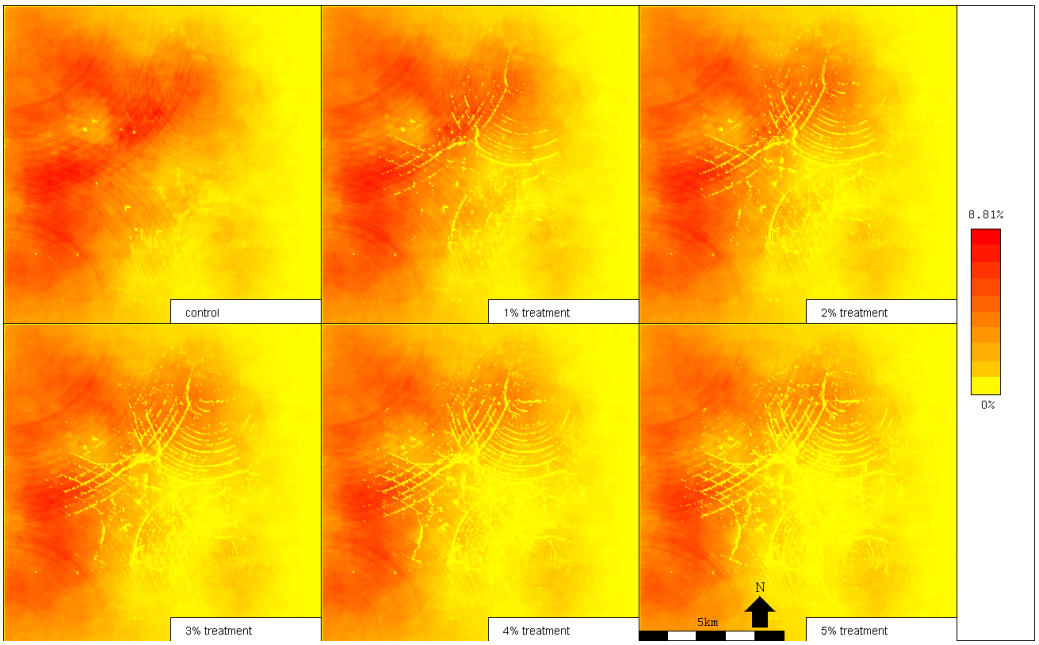
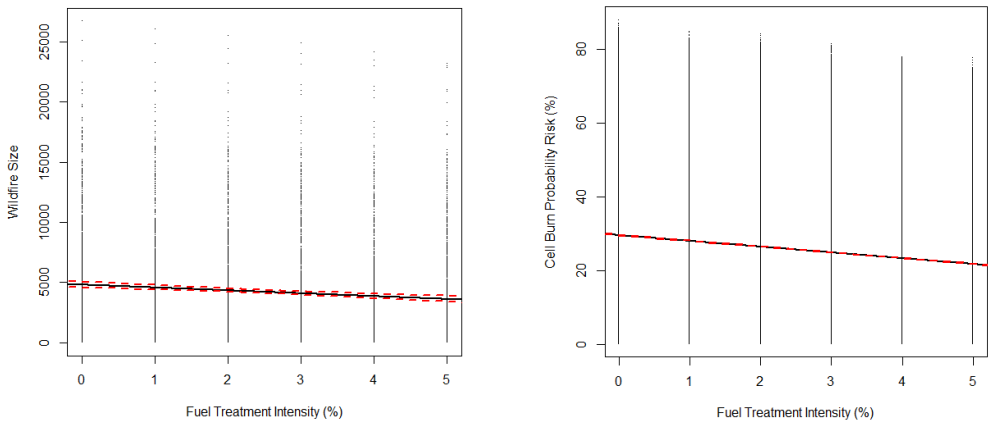


Figure 6: Burn Probabilities Changes

significant (p-value<0.05) in lowering the fire sizes and BP, none of the models had good fits (R^2<0.2). Two post-hoc tests were conducted for model robustness, root-mean-square deviation and mean absolute error. The results indicate that the model was not a good fit, and that although the fuel intervention was statistically significant, it still was not effective in explaining change in both burn probabilities and fire size.

Table 1: Regression Analysis Post-hoc tests

	Sample	R	RMSE	MAE
Burn Probability	Training	0.016	20.883	18.497
	Testing	0.016	20.874	18.471
Fire Size	Training	0.008	4349.0	3385.7
	Testing	0.019	4455.4	3452.8



(a) Fire Size
$\beta = -239.89$ $R^2 = 0.0163$

(b) Burn Probabilites
$\beta = -1.573$ $R^2 = 0.0087$

Figure 7: Regression plots for fire sizes and burn probabilities showing red 95% confidence interval bands, fuel treatment intensity is significant for both variables (p-value$<$0.05)

5 Discussion

The CA model failed to conclusively determine whether or not Finney's minimum travel time was an effective metric by which to conduct fuel treatment optimization in passive forest fire suppression. There is definitely a significant impact, as was statistically and qualitatively shown (Table 1, Figures 3, 5 and 6), however the linear regression models utilized yielded poor fitting results with the post-hoc tests indicating this was not statistically robust.

The model produced results that indicate the MTT fuel treatment intervention suppressed the spread of simulated fires. The decline in fire sizes, fire path lengths and burn probabilities across the board, as shown in Figure 3, show the fuel treatments usually taking effect in the 2%–3% range of treatments. Using most traveled burn paths concentrated the fire suppression around the topography, as most of the fire paths being targeted for optimization are along the elevation changes in the landscape. This makes sense as fire moves faster uphill than down. As seen in Figure

5 and 6, the suppression of fire along the burn paths also has a noticeable effect on the surrounding cells, as the BP of these regions declined by several percent, and by 4% treatment, the whole landscape saw declines in BP. However, the large jump in BP declines between 2% and 3% treatments, and the subsequent slowing down of change between 3% and 5% indicate that the fuel optimization treatments have diminishing returns. Likewise, the effectiveness of the fuel treatments may only really be effective for the largest of fires, as 2% treatments seem to have a strong effect on the largest 10% of fires (Figure 3).

As the regression analysis showed, this method of fire suppression through optimizing fuel treatments by way of MTT fire paths is not a robust method statistically. The model failed the statistical assumptions for a strong fit, and therefore can not be used to explain change in BP or fire size through fuel treatment intervention. It may be the case that the fuel treatments, as detailed earlier, may actually have diminishing returns of effectiveness and that therefore using Finney's MTT paths as a metric for fuel treatments may not actually not be an effective method for targeting fire suppression. Ager et al. (2007) utilized such MTT algorithmic methods to optimize fuel placements and fire suppression. Their method did not use MTT to directly target sources of fuel treatment optimization, rather it was used to simulate the growth of the fire itself. Instead, they used a strand density index (SDI) threshold to thin simulated fuel sources. This seemed to work more robustly, as they found up to 44% declines in BP across their simulation landscapes (Ager et al. 2007). The problem with utilizing MTT directly for fuel treatments is that by removing the most traveled paths for fires, the CA model redirects the fires around the targeted cells rather than suppress the velocity at which the fire advances, like placing a boulder in a fast moving river. This results in small declines in burn probabilities and fire sizes with diminishing returns for fuel treatment interventions. Instead, changing the types of fuel sources the fire can move through to a predominantly lower flammability may yield more robust fire suppression. The CA model could also be improved upon — the wind velocity and duration variables could be fixed to lower the variability in fire sizes,

and the application of a toroidal scenario instead of bounded scenario may allow for spatially non-normal ignition to occur, raising the average size of fires rather than clog the analysis with small fire sizes and ignitions.

6 Conclusion

The utilization of Mark Finney's (2002) minimum travel time algorithm to extract fire paths for fuel treatment optimization in fire suppression may not actually be the most effective means of combating forest fires. Thinning of fuel sources, as had been done in Ager et al. (2007) has proven to be a more robust method of fire suppression, although the use of MTT for fuel optimization may still be effective if targeted fuel thinning around the most burned paths were undertaken.

7 Acknowledgements

Benjamin, Alasdair and Mátyás all had an active engagement in the process of developing and executing the coding for this study. Mátyás guided us through the process of executing CA modeling and was indispensable to the effort. Break down of our contributions is as follows:

Stage	Main	Advice	Language
Cellular Automata Modeling	Mátyás	Alasdair, Ben	Java
Regression Analysis	Alasdair	Ben, Mátyás	R
Plotting, Graphing	Ben	Alasdair, Mátyás	Python

*The R and Python code are located in the appendix, however the Java code was too long to include. You can find the code on github at https://github.com/bench4ng/forestfire-ca-model

References

Ager, A. A., Finney, M. A., Kerns, B. K. & Maffei, H. (2007), 'Modeling wildfire risk to northern spotted owl (strix occidentalis caurina) habitat in central oregon, usa', *Forest Ecology and Management* **246**(1), 45–56.

Bonan, G. B. (2008), 'Forests and climate change: forcings, feedbacks, and the climate benefits of forests', *science* **320**(5882), 1444–1449.

Clarke, K. C., Brass, J. A. & Riggan, P. J. (1994), 'A cellular automaton model of wildfire propagation and extinction'.

Dale, V. H., Joyce, L. A., McNulty, S., Neilson, R. P., Ayres, M. P., Flannigan, M. D., Hanson, P. J., Irland, L. C., Lugo, A. E., Peterson, C. J. et al. (2001), 'Climate change and forest disturbances: climate change can affect forests by altering the frequency, intensity, duration, and timing of fire, drought, introduced species, insect and pathogen outbreaks, hurricanes, windstorms, ice storms, or landslides', *BioScience* **51**(9), 723–734.

Davidson, E. A. & Janssens, I. A. (2006), 'Temperature sensitivity of soil carbon decomposition and feedbacks to climate change', *Nature* **440**(7081), 165–173.

Encinas, L. H., White, S. H., del Rey, A. M. & Sanchez, G. R. (2007), 'Modelling forest fire spread using hexagonal cellular automata', *Applied mathematical modelling* **31**(6), 1213–1227.

Finney, M. A. (2002), 'Fire growth using minimum travel time methods', *Canadian Journal of Forest Research* **32**(8), 1420–1424.

Finney, M. A. (2006), 'An overview of flammap fire modeling capabilities'.

Flannigan, M. D., Stocks, B. J. & Wotton, B. M. (2000), 'Climate change and forest fires', *Science of the total environment* **262**(3), 221–229.

Karafyllidis, I. & Thanailakis, A. (1997), 'A model for predicting forest fire spreading using cellular automata', *Ecological Modelling* **99**(1), 87–97.

Keane, R. E., Drury, S. A., Karau, E. C., Hessburg, P. F. & Reynolds, K. M. (2010), 'A method for mapping fire hazard and risk across multiple scales and its application in fire management', *Ecological Modelling* **221**(1), 2–18.

Kurz, W. A., Dymond, C., Stinson, G., Rampley, G., Neilson, E., Carroll, A., Ebata, T. & Safranyik, L. (2008), 'Mountain pine beetle and forest carbon feedback to climate change', *Nature* **452**(7190), 987–990.

Lindner, M., Maroschek, M., Netherer, S., Kremer, A., Barbati, A., Garcia-Gonzalo, J., Seidl, R., Delzon, S., Corona, P., Kolström, M. et al. (2010), 'Climate change impacts, adaptive capacity, and vulnerability of european forest ecosystems', *Forest Ecology and Management* **259**(4), 698–709.

Millar, C. I., Stephenson, N. L. & Stephens, S. L. (2007), 'Climate change and forests of the future: managing in the face of uncertainty', *Ecological applications* **17**(8), 2145–2151.

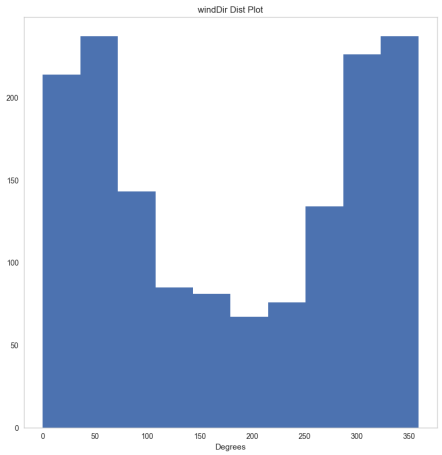
Santilli, M., Moutinho, P., Schwartzman, S., Nepstad, D., Curran, L. & Nobre, C. (2005), 'Tropical deforestation and the kyoto protocol', *Climatic Change* **71**(3), 267–276.

Stocks, B. J., Fosberg, M., Lynham, T., Mearns, L., Wotton, B., Yang, Q., Jin, J., Lawrence, K., Hartley, G., Mason, J. et al. (1998), 'Climate change and forest fire potential in russian and canadian boreal forests', *Climatic change* **38**(1), 1–13.

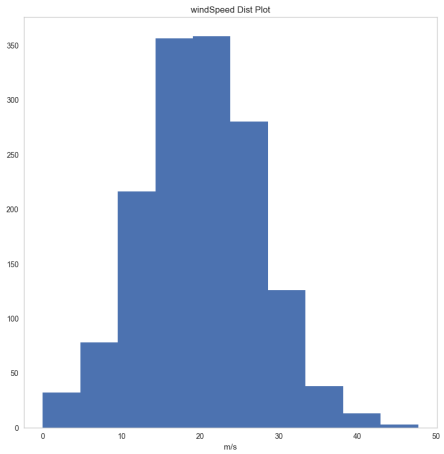
Wolfram, S. (1983), 'Statistical mechanics of cellular automata', *Reviews of modern physics* **55**(3), 601.

Wolfram, S. et al. (1984), 'Cellular automata as models of complexity', *Nature* **311**(5985), 419–424.

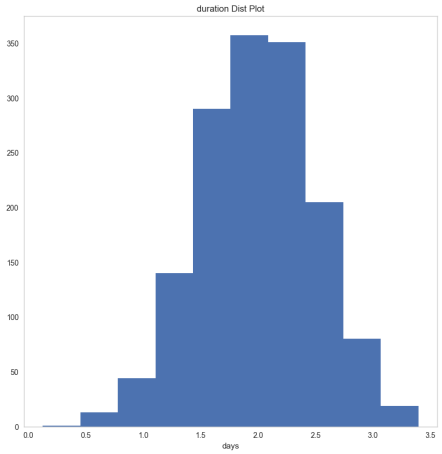
Appendix



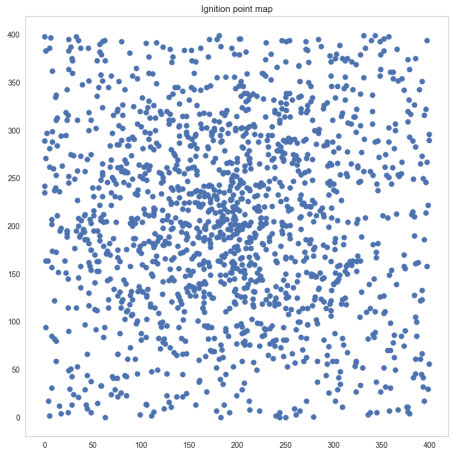
(a) Wind Direction Distribution



(b) Wind Speed Distribution



(c) Duration Distribution



(d) Ignition Point Locations

Figure 8: CA Model Input Parameter Distributions

7.0.1 Python 2

```
import os
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import seaborn as sns
```

```
sns.set_style("whitegrid", {'axes.grid' : False})

print os.getcwd()


if os.path.isdir('outputs') is not True:
    print("Creating 'outputs' directory for images.")
    os.mkdir('outputs')


def mod_diagnostics(model, data, identifier):

    fitted = model.fit()
    dep = model.endog_names
    indep_names = ""

    # create a string containing list of indep names for
       output files
    for name in model.exog_names[1:]:  # we don't want 0
       element as that is the intercept
        indep_names += "{0}_".format(name)

    # Want to include name of DataFrame in the output
       filename but currently DataFrame does not have a
       name attribute
    # So for now use nobs from fitted
    samplesize = str(int(fitted.nobs))

    f1 = open(os.path.join('outputs/ols_outputs', "
       {0}-{1}-{2}OLS_Sample{3}_Summary.txt".format\
        (identifier, dep, indep_names, samplesize)), "w")
    f1.write(fitted.summary().as_text())
    f1.close()

    # calculate standardized residuals ourselves
    fitted_sr = (fitted.resid / np.std(fitted.resid))

    # Histogram of residuals
    ax = plt.hist(fitted.resid)
```

```
    plt.xlabel('Residuals')
    plt.savefig(os.path.join('outputs/ols_outputs', '
        {0}-{1}-{2}OLS-Sample{3}_ResidHist.png'.format\
         (identifier, dep, indep_names, samplesize)),
            bbox_inches='tight')
    plt.close()

    # standardized residuals vs fitted values
    ax = plt.plot(fitted.fittedvalues, fitted_sr, 'bo')
    plt.axhline(linestyle='dashed', c='black')
    plt.xlabel('Fitted Values')
    plt.ylabel('Standardized Residuals')
    plt.savefig(os.path.join('outputs/ols_outputs', '
        {0}-{1}-{2}OLS-Sample{3}_StdResid.png'.format\
         (identifier, dep, indep_names, samplesize)),
            bbox_inches='tight')
    plt.close()


    if len(model.exog_names) == 2:  # univariate model (
        with intercept)

         indep = model.exog_names[1]

         # scatter plot with regression line
         ax = plt.plot(data[indep], data[dep], 'bo')
         x = np.arange(data[indep].min(), data[indep].max
            (), 0.1)  # list of values to plot the
            regression line using
         plt.plot(x, fitted.params[1] * x + fitted.params
            [0], '-',
                  c='black')  # plot a line using the
                      standard equation with parms from the
                       model

         plt.xlabel(indep)
         plt.ylabel(dep)
         plt.savefig(os.path.join('outputs/ols_outputs', '
            {0}-{1}-{2}OLS_Sample{3}_Regression.png'.
```

```
            format\
             (identifier, dep, indep, samplesize)),
                bbox_inches='tight')
        plt.close()


# CONTROL DATA ANALYSIS (DIDNT REALLY WORK...)
df = pd.read_csv("input2.csv")

# print df.head()

ct = df[["windDir", "windSpeed", "duration"]]

# SEABORN DISTPLOTS FOR CONTROL
for i in ct:
    fig = plt.figure(figsize=(10,10))
    fig = plt.hist(ct[i])
    plt.title(i + " Dist Plot")
    if i == "windDir":
        plt.xlabel('Degrees')
    elif i == "windSpeed":
        plt.xlabel("m/s")
    elif i == "duration":
        plt.xlabel('days')
    plt.savefig(os.path.join('outputs', '{0}-Distribution
       .png'.format(i)), bbox_inches='tight')
    plt.close()

fig = plt.figure(figsize=(10,10))
fig = plt.scatter(df["ignX"], df["ignY"])
plt.title("Ignition point map")
plt.savefig(os.path.join('outputs', "ignition-point-map.
   png"), bbox_inches='tight')
plt.close()

df2 = pd.read_csv("descriptives.csv")
print df2.head()

bp = df2[["avgBP", "avgTop10pctBP", "avgFirePathLength"]]
```

```
fs = df2[["avgFireSize", "avgTop10pctFireSize", "
    maxFireSize"]]

# for i in bp:
#     plt.plot(bp[i])
#     plt.show()

# for i in fs:
#     plt.plot(fs[i])
#     plt.show()


f, (ax1, ax2, ax3) = plt.subplots(3, sharex=True, figsize
    =(5, 10))
ax1.plot(df2['intervention'], bp["avgBP"])
ax1.set_title("Mean BP")
ax1.set_ylim([min(bp["avgBP"])-(0.05*min(bp["avgBP"])),
    max(bp["avgBP"]) + (0.05*max(bp["avgBP"]))])
ax2.plot(df2['intervention'], bp["avgTop10pctBP"])
ax2.set_title("Mean Top 10% BP")
ax2.set_ylabel("# Cells")
ax2.set_ylim([min(bp["avgTop10pctBP"])-(0.05*min(bp["
    avgTop10pctBP"])), max(bp["avgTop10pctBP"]) + (0.05*
    max(bp["avgTop10pctBP"]))])
ax3.plot(df2['intervention'], bp["avgFirePathLength"])
ax3.set_title("Mean Fire Path Length")
ax3.set_ylim([min(bp["avgFirePathLength"])-(0.05*min(bp["
    avgFirePathLength"])), max(bp["avgFirePathLength"]) +
    (0.05*max(bp["avgFirePathLength"]))])

f.subplots_adjust(hspace=0.25)
plt.setp([a.get_xticklabels() for a in f.axes[:-1]],
    visible=False)
plt.xlabel("% Intervention")
plt.savefig(os.path.join('outputs', "bp-linegraph.png"),
    bbox_inches='tight')

f, (ax1, ax2, ax3) = plt.subplots(3, sharex=True, figsize
    =(5, 10))
```

```
ax1.plot(df2['intervention'], fs["avgFireSize"])
ax1.set_title("Mean Fire Size")
ax1.set_ylim([min(fs["avgFireSize"])-(0.05*min(fs["
    avgFireSize"])), max(fs["avgFireSize"]) + (0.05*max(fs
    ["avgFireSize"]))])
ax2.plot(df2['intervention'], fs["avgTop10pctFireSize"])
ax2.set_title("Mean Top 10% Fire Size")
ax2.set_ylabel("# Cells")
ax2.set_ylim([min(fs["avgTop10pctFireSize"])-(0.05*min(fs
    ["avgTop10pctFireSize"])), max(fs["avgTop10pctFireSize
    "]) + (0.05*max(fs["avgTop10pctFireSize"]))])
ax3.plot(df2['intervention'], fs["maxFireSize"])
ax3.set_title("Max Fire Size")
ax3.set_ylim([min(fs["maxFireSize"])-(0.05*min(fs["
    maxFireSize"])), max(fs["maxFireSize"]) + (0.05*max(fs
    ["maxFireSize"]))])

f.subplots_adjust(hspace=0.25)
plt.setp([a.get_xticklabels() for a in f.axes[:-1]],
    visible=False)
plt.xlabel("% Intervention")
plt.savefig(os.path.join('outputs', "fs-linegraph.png"),
    bbox_inches='tight')
```

7.0.2 R

```
setwd("C:/Users/Alasdair/Desktop/GSA")

df <- read.csv("[file_name].csv", header = T)
head(df)
summary(df)

# create training and testing sample
set.seed(2017)
train.size <- 0.8
df.train.index <- sample.int(length(df$risk), round(
    length(df$risk) * train.size))
df.train.sample <- df[df.train.index,]
df.test.sample <- df[-train.size,]
```

```
# calculate regression model
fit1 <- lm(risk ~ treatment, data = df.train.sample)

summary(fit1)

# Plot regression model
par(mar=c(5,5,4,2))
regression.plot(fit1, pch=".", conf.bands = TRUE,
                xlab='Fuel Treatment Intensity (%)',
                ylab='Wildfire Risk Probability')

# Create prediction column for both samples
df.train.sample$pred.risk <- predict(fit1, newdata =
   subset(df.train.sample.clean, select=treatment))
df.test.sample$pred.risk <- predict(fit1, newdata =
   subset(df.test.sample.clean, select=treatment))

# Post-Hoc testing on training sample
train.corr <- cor(df.train.sample$pred.risk, df.train.
   sample$risk)
train.RMSE <- sqrt(mean((df.train.sample$pred.risk - df.
   train.sample$risk)^2))
train.MAE <- mean(abs(df.train.sample$pred.risk - df.
   train.sample$risk))
c(train.corr^2, train.RMSE, train.MAE)

# Post-hoc testing on testing sample
test.corr <- cor(df.test.sample$pred.risk, df.train.
   sample$risk)
test.RMSE <- sqrt(mean((df.test.sample$pred.risk - df.
   test.sample$risk)^2))
test.MAE <- mean(abs(df.test.sample$pred.risk - df.test.
   sample$risk))
c(test.corr^2, test.RMSE, test.MAE)
```