A STOCHASTIC APPROACH TO MODELING SURFACE WILDFIRE PROPAGATION

A Dissertation Presented

by

Mohammad Hajian

to

The Department of Mechanical and Industrial Engineering

In partial fulfillment of the requirements for the degree of

Doctor of Philosophy

In the field of

Industrial Engineering

Northeastern University
Boston, Massachusetts

May 2016
ACKNOWLEDGEMENTS

It is a pleasure for me to thank all of those who supported me in any respect during my doctoral studies. I feel truly blessed to have so many wonderful people in my life.

First and foremost, I would like to express my sincere gratitude to my immediate supervisor, Professor Emanuel Melachrinoudis, for encouraging me to get into the PhD program, for his support, continuous guidance and his inexhaustible patience during my PhD study.

I express my special thanks to my Co-advisor, Professor Peter Kubat, for his continuous support and sincere feedback during the course of this study. I would also like to thank Professor Jacqueline Griffin for agreeing to join the dissertation committee and for her valuable comments.

I would like to express my gratitude to Professor James M. Smith and his doctorate student Alexander Smith for their mentorship and for providing the data for this research, without which this work could not be completed.

This journey would not have been possible without the dedicated support of my family, specially my Mom and Dad, my sister and my brother, who gave unstinting support when I needed it most. I would also like to extend my
greatest gratitude to my future in-laws for providing a loving environment for me.

My greatest thanks go to my fiancé, Meena, who always believed in me and has always been my best friend, my faithful partner and my loving and encouraging companion. This accomplishment would not have been possible without her incredible support and encouragement.

I also want to thank my amazing friends who supported me with their uplifting with humorous support and made this journey smooth and enjoyable.

At last I wish to thank many other people whose names are not mentioned here but this does not mean that I have forgotten their help.
TABLE OF CONTENTS

LIST OF FIGURES ............................................................................................................ vii
LIST OF TABLES ................................................................................................................ x
ABSTRACT ........................................................................................................................ xi

1. Introduction .................................................................................................................... 1
   1.1 Overview ..................................................................................................................... 1
   1.2 Motivation .................................................................................................................. 2
   1.3 Proposed research ..................................................................................................... 3
   1.4 Contribution ............................................................................................................. 5
   1.5 Dissertation organization ......................................................................................... 5

2. Literature Review .......................................................................................................... 7
   2.1 How fire spreads ....................................................................................................... 7
   2.2 Modeling the spread of wildfire .............................................................................. 9
      2.2.1 Physical models ................................................................................................. 10
      2.2.2 Empirical models ............................................................................................. 13
      2.2.3 Simulators and mathematical analogous models ............................................ 16
   2.3 The shortest path problem ...................................................................................... 24

3. Wildfire and its behavior ............................................................................................. 35
   3.1 Introduction ............................................................................................................. 35
   3.2 Rothermel’s fire model and BEHAVE ................................................................. 36
   3.3 Wind distribution .................................................................................................... 41
3.4 Fire speed distribution.................................................................44
3.5 Wind distribution from real data....................................................47
4. Network construction ..................................................................51
  4.1 Network nodes ..........................................................................53
  4.2 Delaunay triangulation...............................................................55
  4.3 Edge directions ..........................................................................57
  4.4 Fire ROS ....................................................................................58
  4.5 The study area and the real-sized network ...............................62
    4.5.1 Why shortest path? ..............................................................66
5. Wildfire front-line stochastic contours ........................................67
  5.1 Introduction ...............................................................................67
  5.2 Problem statement ......................................................................68
  5.3 Depth-Limited Dijkstra (DLD) ....................................................69
  5.4 Contour construction .................................................................72
  5.5 Monte-Carlo implementation .....................................................73
  5.6 Contour integration .................................................................74
  5.7 Stochastic evacuation contours ................................................75
  5.8 Experimental results .................................................................76
  5.9 Conclusion .................................................................................78
6. Distribution of the shortest fire travel time ...................................80
  6.1 Introduction ...............................................................................80
  6.2 Fire travel time ..........................................................................82
  6.3 Network size reduction ..............................................................84
    6.3.1 Dominance criterion ............................................................85
    6.3.2 Network cut-down step 1: node elimination .......................92
6.3.3 Network cut-down step 2: path enumeration ........................................... 98
6.4 Shortest time histogram ............................................................................. 105
  6.4.1 Monte-Carlo simulation ......................................................................... 105
6.5 Evaluation: Original vs. reduced-size network .......................................... 108
6.6 Experimental results .................................................................................... 110
  6.6.1 Network pruning .................................................................................... 112
  6.6.2 Distribution of the fire traversal time ...................................................... 117
6.7 Value of the reduction methodology ............................................................ 120
6.8 Multiple ignition points ................................................................................ 121
6.9 Stochastic versus deterministic analysis ....................................................... 123
6.10 Conclusion ................................................................................................... 126
7. Summary and future research ....................................................................... 129
REFERENCES ................................................................................................. 132
LIST OF FIGURES

Figure 1: The Fire Triangle. Fuel, heat, and oxygen are the required ingredients for fire ........................................................................................................................................ 8
Figure 2: Position of the fire after a certain time with uniform wind in x direction ........................................................................................................................................ 17
Figure 3: The Outer surface of all the ellipses becomes the new fire perimeter ... 17
Figure 4: No-wind fire .......................................................................................................................................................................................... 40
Figure 5: Wind-driven fire .................................................................................................................................................................................. 40
Figure 6: Upslope fire (Rothermel, 1972) .................................................................................................................................................. 40
Figure 7: Fixed $\lambda$, varying $k$ .................................................................................................................................................. 44
Figure 8: Fixed $k$, varying $\lambda$ .................................................................................................................................................. 44
Figure 9: Wind speed histogram for real data ........................................................................................................................................ 49
Figure 10: Wind speed cumulative histogram for real data .................................................................................................................. 49
Figure 11: Linear regression least square line ........................................................................................................................................ 50
Figure 12: Weibull distribution fit to the data histogram .................................................................................................................. 50
Figure 13: Overlaying of fuel and height layers ........................................................................................................................................ 54
Figure 14: Inside point for regions .................................................................................................................................................. 54
Figure 15: Delaunay triangulation and Voronoi diagram of the example area.... 56
Figure 16: Voronoi diagram is the dual of Delaunay triangulation .................................................................................................................. 56
Figure 17: Comprising parts of Delaunay triangulated edges .................................................................................................................. 57
Figure 18: Azimuth ...................................................................................................................................................................................... 59
Figure 19: Effect of wind on eccentricity ................................................................. 60
Figure 20: Fire spread calculations for edges......................................................... 61
Figure 21: Possible projected fire speed ................................................................. 62
Figure 22: Montague city in Franklin County, MA (Wikipedia)............................. 63
Figure 23: Montague Wildfire Management Area (Google, 2015)......................... 63
Figure 24: Network representation of MPWMA ..................................................... 65
Figure 25: Radial sweeping to construct the contours from contour points ........ 72
Figure 26: Contour integration ............................................................................. 74
Figure 27: Stochastic contours \( t=43 \) min. ............................................................ 77
Figure 28: Dominance criterion ........................................................................... 86
Figure 29: Dominance criterion for dependent paths ......................................... 89
Figure 30: Worst-case dominance error ............................................................... 90
Figure 31: Coefficient of variation versus mean for randomly selected paths..... 94
Figure 32: \( k \)-shortest path in a planar triangulated graph ................................. 99
Figure 33: Minimal area with two components .................................................... 101
Figure 34: Finding the minimal area .................................................................... 104
Figure 35: Study area network with selected source and destination nodes ...... 112
Figure 36: Effect of wind speed on step 1 output nodes ..................................... 113
Figure 37: The effect of wind direction explained using the ellipse model for wind directions: (a) 0, (b) 90, (c) 180 and (d) 270....................................................... 114
Figure 38: Output networks depicting the nodes from node elimination step as well as the generated paths from the path enumeration step for wind directions: (a) 0, (b) 90, (c) 180 and (d) 270................................................................. 115
Figure 39: The shape of the restrained area for wind directions: (a) 0, (b) 90, (c) 180 and (d) 270....................................................................................... 116
Figure 40: Comparison of the fire arrival time empirical distribution of the reduced-size network versus the distribution derived from the original network. ........................................................................................................................................ 118

Figure 41: Q-Q plot of the fire arrival time empirical distribution of the reduced-size network versus the distribution derived from the original network. ........ 119

Figure 42: Monte-Carlo implementation time on the reduced-size network compared with the implementation time on the original network. ................. 121

Figure 43: Multiple ignition points, multiple destination points and already propagated fires are handled using dummy node replacement......................... 122
LIST OF TABLES

Table 1: Categories of the shortest path problem ............................................................... 26
Table 2: Anderson’s 13 fuel models and their parameters ................................................. 37
Table 3: Important notation used in Rothermel’s equations ............................................. 39
Table 4: Data peek for wind speed ..................................................................................... 48
Table 5: Sample attribute table for the network edges .................................................... 59
Table 6: Fire pattern versus Dijkstra’s shortest path ......................................................... 66
Table 7: Calculated mean and standard deviation for edges ............................................. 111
Table 8: KS-Statistic values for the four scenarios compared with the critical value .......... 120
Table 9: Effect on wind direction on the number of Step 1 nodes ............................... 120
ABSTRACT

Wildfires have significant local economic and social effects. On average, more than 100,000 wildfires clear 4 million to 5 million acres (1.6 million to 2 million hectares) of land in the U.S. every year (2014). Developing effective analytical models to accurately predict the wildfire propagation enables us to take preventive measures in a timely manner. In this research, a stochastic methodology for modeling surface wildfire propagation is presented. First, Delaunay triangulation is employed to construct a directed network representing the landscape where fire is assumed to travel along the edges of the network. Wind speed variability is then represented as a continuous random variable. The distribution of the fire rate of spread is derived consequently using Rothermel’s equations and the wind speed distribution.

Two problems are studied in this research. In the first problem, we introduce the concept of wildfire stochastic contours, which represent the fire front-line state. The contours are derived in a way that enable the decision maker to analyze the confidence of the prediction. In the second problem, a methodology to estimate the fire traversal time distribution from a source to a point of interest is presented. This methodology takes advantage of a two-step network reduction module to boil down the network and deduct a computationally tractable one.
Monte-Carlo simulation is then utilized to estimate the fire arrival time distribution to the point of interest on the reduced-size network. Finally, the contour generating model and the distribution estimator are both implemented on a study area to demonstrate the effectiveness of the models in a real-world problem. The results from the experiment are then presented and evaluated for accuracy.
1. Introduction

1.1 Overview

According to Cambridge dictionary a wildfire, also known as a wildland fire, forest fire or vegetation fire, is an uncontrolled fire mostly occurring in wildland areas. Wildfires often begin unnoticed, but they spread quickly. A wildfire differs from other fire types by its extensive size, the speed at which it can spread out from the ignition point, its ability to change direction unexpectedly, and the ability to jump gaps such as roads and rivers (NIFC, 2014).

Several factors can affect the intensity and direction of a wildfire. Among them, some are slow changing and almost static during the wildfire propagation time. Terrain and the fuel distribution of the landscape are example of such factors. Some, on the other hand, are fast changing and very dynamic such as weather conditions. Wind, among the other weather conditions, is prone to sudden and quick changes.

Containing the fire and evacuating the in-danger population are two of the common strategies of fighting wildfires in emergency events. However, without the knowledge of wildfire whereabouts, none of the abovementioned strategies can be executed properly. In order to predict the wildfire whereabouts one needs to predict the fire rate of spread and therefore to predict all the factors involved.
in spreading the fire. However, factors such as wind speed are hard to predict and thus make this prediction challenging.

In this research we provide a stochastic framework to predict the fire rate of spread (ROS) by considering the variability of the wind speed. Two prediction models are presented, one for predicting the location of the fire front-line and one for estimating the fire traversal time distribution through the landscape.

1.2 Motivation

Wildfires can have significant local economic effects, both short-term and long-term, with larger fires generally having longer-term and greater impacts. Wildfires may directly restrain recreation and tourism in and near the fires. Extensive fire damage to trees can significantly alter the timber supply. Water supplies can be degraded by post-fire erosion. If an area’s aesthetics are impaired, local property values can decline.

Looking at the statistics, on average more than 100,000 wildfires clear 4 million to 5 million acres (1.6 million to 2 million hectares) of land in the U.S. every year (National-Geographic, 2014). Forests are not the only places affected by wildfires. The amount of pollution going to the atmosphere by large wildfires can cause significant ecological and climatic implications. A wildfire can move at speeds up to 14 miles an hour (23 kilometers an hour), consuming everything in its path (National-Geographic, 2014). Wetting, fuel reduction, creating fire breaks and control lines are some of the techniques being used to reduce the fire damage.
Predicting the fire rate of spread and the location of the fire front-line can greatly help in choosing the best fighting strategy and preparing the in-danger areas before the fire arrives. An incorrect estimate of the fire location and propagation speed could lead to catastrophic results. This emphasizes the need for accurate and reliable fire prediction tools.

In this research, we develop a stochastic fire spread model with the aim to predict the wildfire spread more accurately and realistically. The model helps the decision maker in adopting the best action plan by providing a confidence level of the predictions, which allows for better evaluation of the risk involved in the decision making process.

1.3 Proposed research

The purpose of this research is to provide a framework for predicting wildfire propagation over a heterogeneous landscape. The work here is originated from the idea by Stepanov and Smith (2012) in which the landscape was modeled as a planar graph. In their work, a deterministic model for predicting the fire front-lines was presented.

Several factors have influence on the direction, intensity and spread rate of the wildfire. Some of the factors are fast changing and unpredictable and thus a purely deterministic model may fail in providing a reliable prediction for the fire propagation. In this research, we provide a stochastic framework for modeling wildfire propagation that describes the variability of the wind speed, an important factor in determining the fire rate of spread, by the mean of random variables. A discrete Delaunay representation of the network is used to model
the landscape and fire spread paths using a network, an idea by Stepanov and Smith (2012). In this representation, fire is assumed to traverse along the edges of the network.

With the wind speed described by a random variable, the fire rate of spread and thus the fire traversal times along the edges of the network will also be random variables as they are functions of wind speed. This function can be derived from the well-known Rothermel’s equations (Rothermel, 1972). Two models are then presented with the following objectives:

1) Distribution of the fire traversal time from its ignition point to the point of interest.

2) Wildfire front-line stochastic contours.

For the first objective, a fast simulation methodology to estimate the distribution of the fire arrival to the point of interest is presented. In this model, a two-step reduction methodology is introduced with the goal of reducing the run-time of the simulation. This methodology attains this goal by cutting down the parts of the network that have negligible contribution to the estimation error of the distribution. Then, a Monte-Carlo implementation of the shortest path Dijkstra’s algorithm is utilized to estimate the fire traversal time distribution.

For the second objective, the predicted fire front-line location is represented using stochastic fire contours or “stochastic contours”. A fire contour is an outline of the fire front-line enclosing the burnt area. The state of fire at time $t$ given the ignition point of time 0 can be shown by its contour. As the fire rate of spread in our model is a random variable, the fire contours are also of stochastic nature. Thus, the state of fire front-line at time $t$ is depicted using multiple
stochastic contours, each representing a different confidence level for the prediction. A decision maker can use the model to generate the predicted fire contours based on his/her desired confidence level.

1.4 Contribution

There are two main contributions in this research. First, to the best of our knowledge, this research is the first study on modeling the fire front-lines stochastically with the model’s ability to consider the variability of the wind. For this purpose, first a revised version of the Dijkstra’s algorithm is presented to employ in the subsequent simulation model and then a methodology to integrate the simulation results is proposed.

Second, a two-step reduction methodology is introduced to reduce the size of the network representing the landscape as a pre-step for the subsequent simulation. The methodology is able to efficiently reduce the run-time of the simulation without significant effect on the accuracy of the results. An empirical distribution of the fire arrival time to the point of interest is derived as the output of simulation, making our model the first to practically provide the fire traversal time distribution.

1.5 Dissertation organization

In Chapter 2, we go over the previous studies in the area of fire propagation modeling and also the shortest path problem. First, the fire models are classified into three categories, starting from models that merely consider the fundamental
physical and chemical fire reactions to models that deal with the general behavior and shape of the fire. Then, we look at the literature of the shortest path problem and how it can be utilized in the context of wildfire. Following will be Chapter 3, which focuses on investigating the factors that affect the wildfire behavior. We discuss the use of random variables to model the wind speed and how to derive the distribution of the fire rate of spread using the wind speed distribution. The use of Rothermel’s equations in deriving the distribution is explained in detail. In Chapter 4, we explain the process of representing the fire environment as a network using Delaunay triangulation and GIS functions. Also, we go over the study area, which will be utilized for our experiments in the subsequent Chapters. In Chapter 5, we introduce the concept of wildfire stochastic contours and wildfire evacuation contours. We develop the Depth-Limited Dijsktra’s algorithm to construct the fire contours, which is used in a Monte-Carlo model to derive the stochastic contours. In Chapter 6, we discuss the problem of finding the distribution of the fire traversal time from its ignition point to a point of interest. We introduce a novel two-step network size reduction methodology, which is utilized to speed up the subsequent simulation module that is used to estimate the abovementioned distribution. At the end of Chapter 6, experimental results are reported and the accuracy of the method is evaluated. Finally, in Chapter 7, we summarize the results of this study and discuss the potential improvements and the future work of the research.
2. Literature Review

2.1 How fire spreads

According to NFPA 921 (Association & Association, 2008) fire is "a rapid oxidation process, which is a chemical reaction resulting in the evolution of light and heat in varying intensities". There are three necessary ingredients for the occurrence of the fire: vegetation, oxygen and a heat source. These ingredients are commonly illustrated by the fire triangle model (Figure 1). Vegetation is the combustible source for the chemical reaction. Oxygen acts as an oxidizing agent and a heat source initiates and sustains the chemical reaction (Pyne, Andrews, & Laven, 1996). When there is not enough generated heat or when water reduces the heat level or when the fuel is removed, isolated, or exhausted or when there is limited source of oxygen, one edge of the triangle is broken and the fire is extinguished.

The fire gradually consumes the vegetation by spreading across the landscape. This process has four combustion phases: pre-heating, ignition, combustion and extinction (Pyne et al., 1996). The heat released from the fire front has the most intense flaming combustion. This heat is transmitted to the untouched vegetation and heats it up to the ignition temperature (Almeida & Macau, 2011). New flames rise as the result of this process and fire moves to a new position. As long as there is unburnt vegetation, this chemical process continues.

For a more technical explanation, it helps to consider the fire as a series of ignitions. Fire supplies the heat to the potential fuel. This causes the surface to
dehydrate and gradually raises the surface heat until the fuel begins to “pyrolyze”, releasing combustible gases. When the gas evolution rate from the potential fuel is sufficient to support the combustion, it gets ignited by the flames and the fire advances to a new position (Rothermel, 1972).

**Figure 1: The Fire Triangle. Fuel, heat, and oxygen are the required ingredients for fire**

There are three recognized patterns of fire spread on the basis of fuel layers: Ground fire, surface fire and crown fire (Alexander & Cruz, 2011). Ground fires, which burn organic matter in the soil, spread very slowly without any visible flames. Surface fire burns fuels located at ground level such as leaf litter and fallen branches and has a flaming front. Crown fire, also known as canopy fire, is dependent on the surface fire for its spread and needs heavy fuel load and strong winds to continue to burn (Nps.gov, 2014).

The focus of this research is on fire propagation through surface fire. Accidental surface fire is one of the most common threats to tropical forests. Surface fires can have huge impacts on rainforest plants, and can greatly increase the likelihood of far larger fires that can lead to complete destruction of a forest (Laurance, 2003).
2.2 Modeling the spread of wildfire

Although the literature is rich on fire models, most of the works in the area study the fire behavior from an ecological point of view, which fall under the category of physical models as classified by Sullivan (2007c). According to his classification scheme, there are three categories of fire models: 1) physical models, 2) empirical models, and 3) mathematical analogous models and simulators. Physical models are those concerned with the mathematical analysis of the fundamental physical and chemical processes of fire spread. They do not involve a real fire event (Perry, 1998). Empirical models are those dealing with statistical descriptions of wildfires and hence they do not involve the fundamental mechanisms that drive the fire process (Perry, 1998). Another category of fire spread models are mathematical analogous models and simulators. These models focus on the simulation of fire spread across the landscape from a holistic perspective (Sullivan, 2007b). We adopt the same categorization scheme in our review of literature. Our approach here falls within the third category, although it also utilizes an empirical model for its calculations. We aim to predict the wildfire’s general pattern and its final shape rather than quantitatively estimating its behavior. These different models can be regarded as a continuum ranging from those that purely deal with fundamentals of the fire process, to models that are only concerned with providing a way of explaining the fire growth regardless of the processes driving it. Physical models and simulators stand at the opposite ends of this continuum and empirical models take a place in between.
We start by taking a brief look at different physical models and their features. We then continue with a review of the existing empirical models including Rothermel’s model, which will be utilized as a sub-module in our proposed model. Some of the works in the literature are described by their authors as semi-physical or semi-empirical as they neither fall under the category of physical or empirical models by definition. We will classify those in either the physical or empirical categories depending on their main functionality. Finally, in the third category, we review the different mathematical analogous and simulators. In each category, we also take a look at some of the related academic and commercial fire utilities and software tools that were developed based on the corresponding models.

It should be noted that our focus here is on the surface fire models. Surface fire is the “fire that burns in litter and other fuels at or near the surface of the ground, mostly by flaming combustion” (Brown & Smith, 2000).

2.2.1 Physical models

The first physical model for the surface fire spread was developed by Fons (1946). In his model, energy conservation equations were applied to obtain the fire spread versus logarithmic growth of fuel bed. The model outcomes were validated by experimenting on pine needles. Despite some errors and shortcomings, his results were considered by later researchers to be relatively good. A series of physical models emerged between 1960 and 1990. Their approaches to the description of the physical system were almost identical but different in the way that theoretical principles were applied (Pastor, Zarate,

Forbes (1997) developed a simple two-dimensional combustion model for large-scale bushfire spread that takes into account radiative effects, fuel consumption and wind. He utilized a set of differential equations in his model to describe the fire behavior. His findings show that unlike what was claimed before, fully developed bushfires are unlikely to propagate as “periodic travelling waves”, a type of solution for partial differential equations. He pointed out the usefulness of his model in enabling practitioners to decide on how wide a firebreak needs to be.

FIRETEC (Linn, 1997), a fire model based on the principle of momentum, conservation of mass and energy, which was developed at the Los Alamos National Laboratory, is capable of predicting time-dependent fire behavior in three dimensions. FIRETEC relies solely on the formulations of the chemistry and physics and to model the fire behavior (Sullivan, 2007c). FIRETEC uses a simplified process to represent the complex reactions of a wildfire. IUSTI was developed based on a weighting average procedure describing the fire-induced behavior by Larini, Giroud, Porterie, and Loraud (1998). Unlike FIRETEC, IUSTI concentrates on resolving the chemical and conservation equations at a much smaller spatial scale but at the cost of sacrificing the 3-dimensional solutions (Sullivan, 2007c).

Mell, Jenkins, Gould, and Cheney (2007) introduced WFDS, a model developed at the National Institute of Standards and Technology (NIST). WFDS was an extension of the structural Fire Dynamics Simulator (FDS), a computational fluid
dynamics (CFD) model of fire-driven fluid flow. WFDS was a comprehensive model based on a three-dimensional, fully-transient, physics-based approach for modelling surface fire spread. Two experimental cases were simulated in order to evaluate the model’s predictions of the fire perimeter advances. The overall WFDS predictions of the head fire spread rate were demonstrated to be similar to an empirical relation.

Moving towards empirical models, Carrier, Fendell, and Wolff (1991) tried to derive a formula for the fire rate of spread in a specially designed wind tunnel. The effects of parameters such as wood species, fuel-element length, fuel-bed width and moisture content of the fuel on the fire rate of spread were investigated based on the assumption that the fire achieves a “quasi-steady” rate of speed in constant wind speed and fuel conditions. Another “semi-empirical” model for a linear fire-front propagating through homogeneous bed was presented by Vaz, André, and Viegas (2004). A basic case of no wind, no slope, and fine fuel particles was considered in their study. The model was not “closed” meaning that it required two properties of the fire front as inputs. Vaz et al. showed that the model’s predictions are very sensitive to the flame height estimated by an empirical sub-model. If the sub-model was not used and the flame height was entered as an input, the predictions of the model for the rate of spread were within the expected experimental errors.

Among the factors affecting the wildfire propagation, slope is one of the important contributors. The fire rate of spread can increase or decrease considerably on an inclined surface compared to horizontal spread rate. Santoni and Balbi (1998) examined the slope effect on the fire spread across a Pinus pinaster litter. They were able to consistently reproduce the main characteristics
of spreading, and confirmed that the fire spread rate increases with increasing slope values. They demonstrated the relevance and capability of their model with regards to slope changes by comparing their results with experimental observations.

2.2.2 Empirical models

Empirical models are developed from historical wildfire studies or from experiments and observations. Therefore, they are applicable only to the circumstances in which conditions are identical to those existed when deriving the model. The focus of empirical models has been on the key characteristics that describes the fire behavior. These generally are the rate of fire head forward spread (ROS), the height of the flames, their angle and their depth (Sullivan, 2007a). The primary use of such models has been to predict the ROS in the direction of the wind.

One of the first and most notable works in this area is the model developed by Rothermel (1972). His fire spread model predicts the fire behavior and takes into account the mean wind velocity, slope of the terrain, propagating flux and also different aspects of fuel such as fuel depth, fuel heat, fuel surface to area ratio and fuel moisture. He constructed equations that take all the above-mentioned parameters as input and gives an approximation for the fire ROS as output. Rothermel’s fire behavior model consists of about 80 equations. If used to its full extent, it requires 17 input variables (Bachmann & Allgöwer, 2002). Rothermel’s model forms the basis of the National Fire Danger Rating System and the fire prediction tool BEHAVE (Sullivan, 2007a).
As mentioned earlier, the focus of many empirical models is on finding a relationship between the fire behavior and the factors affecting it. Fire behavior can be described best by the fire ROS. In an experiment on Pine pinaster, Fernandes, Botelho, and Loureiro (2002) tried to establish a quantitative relationship between fire behavior and fire environment. In order to do that, fire spread rate was modeled as a function of fuel moisture content, wind speed, air temperature and fuel type. Three alternative models including Rothermel’s were examined and evaluated against the results of their experiment. Performance of these three models for the established environment were shown to be unsatisfactory, underestimating the fire rate of spread, with Rothermel’s having the best predictions among them. Marsden-Smedley and Catchpole (1995) presented another model for predicting the fire ROS and flame height of fires. They found the surface wind speed to be the main factor affecting the fire rate of spread. Dead fuel moisture content and the time since the last fire were identified as the other key factors.

Again, with the objective of calculating the fire rate of spread, Cheney and Gould (1995) developed a model to predict ROS in grasslands using the data from wind speed, dead fuel moisture and the pasture type (natural, grazed or eaten-out). Their study was primarily focused on finding the relative importance of different fuel characteristics on the fire ROS. They were able to develop a model based on the spread rate measurements of experimental fires, which used two case-specific functions to describe the relationship between the wind speed and the fire rate of spread. For the case of a wind speed lower than 5 km/h, the relationship was described by a linear function and for the winds blowing above 5 km/h, the relationship was described by a power function of wind speed.
Marsden-Smedley and Catchpole (1995) focused their study on buttongrass moorland fires and performed an extensive study using different methods for collecting data on site location, fuel loads, fuel moisture, fuel heat content and fuel consumption. Seven models including Rothermel’s were examined for their ability to predict the fire ROS. Performance of each of the models were compared with regard to each of the factors contributing to the fire spread to see which model outperforms the others under what conditions.

In another study by Albini (1996), fire ROS and shape of ignition were predicted given the radiometric temperature of the burning zone by testing on laboratory fires. Radiation is considered by many researchers to be the dominant mechanism of heat transfer driving a fire. Albini’s model was able to accurately predict the spread rate for experimental fires. Focusing on a specific area, mallee-heath shrublands in semi-arid southern Australia, Cruz, McCaw, Anderson, and Gould (2013) used two experimental burning programs to develop models for the sustainability of fire spread, forward spread rate, fire type and flame height. ROS was modelled using nonlinear regression analysis and wind speed and litter fuel moisture content were recognized as the best descriptors for the fire ROS.

Analysis of uncertainty propagation was studied in the work by Bachmann and Allgöwer (2002), who applied first-order Taylor series to Rothermel’s equations. Bachmann and Allgöwer introduced an approximation error that reflects the variation of the input uncertainty compared with the smoothness of an output function. Analysis of uncertainty in fire propagation, which is also the focus of our research, has appeared in only a few studies in the literature, most of them
falling under the mathematical analogous models and simulators. We will refer to some of these models in the review of the third category.

2.2.3 Simulators and mathematical analogous models

Models in this category are distinguished by their holistic view of fire representation. Rather than dealing with the underlying mechanisms, they are focused on describing the system behavior as a whole. Mathematical analogous models are those that utilize a mathematical percept rather than a physical one in describing the system behavior. Simulators are mostly implementations of previously developed models and do not represent a model on their own.

One of the first and most important models in this category is the one developed by Anderson, Catchpole, De Mestre, and Parkes (1982) who first used the concept of Huygen wavelet principle for modeling the fire perimeter propagation. They developed a simple elliptic model for the spread of fire front through grass fuel. The grass fuel was assumed to be of homogeneous supply on a perfectly flat terrain. In their elliptic mode, the shape of the ellipse, which shows the shape of the fire perimeter, is determined by the ROS (Figure 2). Each point on the fire perimeter then becomes a new source of fire, forming a new ellipse. As the time advances, according to the Huygen principle, the outer surface of all the ellipses becomes the new fire perimeter (Figure 3). This new fire perimeter is no longer elliptical, except in the special case where there is no change in wind.
Using the same concept (Huygen wavelet principle) for modeling fire perimeter, Plourde, Doan-Kim, Dumas, and Malet (1997) developed a mathematical algorithm to estimate the propagation of the fire front with time. Instead of the ellipse approach in which the progress of fire is regarded as the propagation of a discrete number of points, they utilized a complex Fourier series function to model the continuous fire-front curve. Although their results were finer and more accurate, the solution required a heavy computational cost to achieve.

The well-known 13 fire behavior fuel models were introduced by Anderson et al. (1982). Fuel models describe the characteristics of certain fuel types that were studied in the past, as input parameters (Rothermel, 1972). Using fuel models eliminates the need for repetitive measurements of previously studied fuel types.
Anderson’s fuel models consisted of 4 fuel groups: grasslands, scrublands, timber, and slash. He cross-referenced the 13 fire behavior fuel models to the 20 fuel models of the National Fire Danger Rating System using a similarity chart.

Capable of consuming Anderson’s fuel models, BEHAVE, a fire behavior prediction and fuel modeling system was introduced in 1986. BEHAVE has two subsystems, FUEL and BURN. BURN and FUEL were developed by Andrews (1986) and Burgan and Rothermel (1984), respectively. According to Burgan and Rothermel (1984), there are four ways to obtain a fuel model in BEHAVE: 1-Select one of the 13 fire behavior models. 2-Modify one of these models. 3-Use the measured data of one of the “inventory” techniques. 4-Assemble a new fuel model using the FUEL subsystem. BURN subsystem of BEHAVE is designed to obtain a fuel model using one of the 4 methods, along with fire prediction techniques for predicting fire behavior.

FARSITE, a widely used fire simulator in the US, is an area simulator for predicting fire spread across the landscape. FARSITE was developed based on BEHAVE (Sullivan, 2007b). FARSITE integrates different surface fire models using a vector propagation technique to predict the fire perimeter expansion, and is able to produce vector fire perimeters at specified time intervals (Mark Arnold Finney, 2004).

Green, Tridgell, and Gill (1990), developed IGNITE, a landscape fire modelling system for fires on heterogeneous fuels. In this modelling system, landscape is represented by rectangular array of “cells”, each cell representing a separate area of the land surface. This rectangular array of cells is often referred to as a raster. Associated with each cell are environmental factors such as vegetation
type and fuel abundance. As the time advances, the status of each cell is updated, and the fire map is revised. Fire spread is then modeled as an epidemic process, assuming that fire spreads across a landscape following the least time-traveled path. In another study using the raster representation, Vasconcelos and Guertin (1992) introduced FIREMAP, a simulation system that simulates the fire growth in discrete time steps. Their simulation system integrates Rothermel's equations with a raster-based geographic information system (GIS) in which data is stored in the form of a grid of cells. The fire ROS for each cell is calculated assuming homogeneous fire environment for that cell.

Cellular Automata (CA), simulation methods based on a grid of cells, are widely used approaches in wildfire behavior modeling. A cellular automation consists of a grid of cells, associated with each a number of “states” such as 0 and 1. For each cell, a set of “neighborhood” cells are defined, which typically are a list of adjacent cells. An initial state is assigned to each cell at time 0. As the time advances, a new state for each cell is calculated according to a rule that is defined based on the state of the current cell and the states of the neighborhood cells. Time advancement is made in discrete numbers. The updating rule for the state of cells typically remains the same throughout the simulation. Karafyllidis and Thanailakis (1997) partitioned a forest landscape into a matrix of identical square cells to implement Cellular Automation. The state of each cell at a certain time was defined as the ratio of the burned area of the cell to its total area. ROS was calculated for each cell based on a) the wind speed and direction, and b) the shape and height of the cell land. The fire front-line was simulated given these inputs and the starting time. In his later paper, Karafyllidis (2004) designed a genetic algorithm to speed up this algorithm, which enabled the model to
implement on parallel processors. With this new design, users were able to examine different fire-spreading scenarios in a timely manner.

Using a different fire growth model, but with the same CA paradigm, a simulation method to model the spatial and temporal behavior of wildfires was proposed by Clarke, Brass, and Riggan (1994). Their simulation was based on a modification of the growth process of diffusion-limited aggregation (Adlakha, 1986). DLA is a process-oriented model that assumes an object’s growth and change of shape to be the result of a large number of random events, each of which influenced by the events preceding it. Fire propagation is a process closely resembling that of DLA.

To avoid the “spurious symmetries” of square cells, which are common in CA models, Trunfio (2004) proposed a hexagonal lattice-based Cellular Automation. Like many other researchers, he based his calculation on the fire spread relationships developed by Rothermel. The state of each cell was composed of sub-states such as altitude, vegetation, temperature, humidity, wind direction and wind speed. The predicted fire developments claimed to have a significantly fit with real data.

Mraz, Zimic, and Virant (1999), incorporated the concept of fuzzy logic in their CA model to predict the wildfire propagation shape. Fuzzy logic enabled their model to use descriptive knowledge of fire behavior given by firemen, which are often approximated from uncertain data. Fuzzy logic words were also utilized to interpret the relationships between fire growth and the factors affecting it. For example, the spread of fire was presumed to get “more or less” accelerated in the wind direction. They mentioned Long implementation time as the only disadvantage of their model compared to deterministic ones.
Taking a probabilistic approach, Hargrove, Gardner, Turner, Romme, and Despain (2000) developed a model called EMBYR to simulate large fires propagation through heterogeneous landscapes. In their model, landscape is depicted as a grid, and the probability of spread from each cell of the grid to eight neighbors is calculated based on fuel type, fuel moisture, wind speed and direction. They defined additional rules for non-neighbor cells as well. “Fire bands” were distributed to non-neighbor downwind cells to calculate the probability of ignition in those areas based on fuel type and moisture. In a different approach but using probabilistic measures, Boychuk, Braun, Kulperger, Krougly, and Stanford (2009) developed a stochastic fire growth model with the aim of predicting the behavior of large forest fires. They modeled the fire as a random phenomenon on a grid of spatial locations. More specifically, their method was based on a continuous-time Markov-chain on a lattice network. Although no actual implementation was provided in their work, they claimed that the model implementation in a computing environment allows one to obtain probability contour plots, burn size distributions, and distributions of time to specified events.

Richards (1995) utilized differential equations to model two-dimensional fire spread for heterogeneous fuel and meteorological conditions. His equations represented the fire perimeter as a closed curve and required the specification of the ROS at each point on the curve. It was shown that the fire ROS is a function of the angle between normal vector and the wind direction vector at each point on the perimeter. In a later work, he extended his model to a more general case forming a mathematical model of fire perimeter growth over a 3-dimensional surface (Richards, 2000).
Looking from a different perspective, Cova, Dennison, Kim, and Moritz (2005) used wildfire spread modeling to determine wildfire evacuation buffer. The evacuation buffer was defined in a way that when the fire crosses the boundaries of the buffer, an evacuation is recommended. The model was based on a cell-type surface, with the rate of spread computed for every cell in eight directions. Fire spread for each cell was calculated using the FLAMMAP software package which is based on the FARSITE system. A network-based representation of fire spread was constructed, associating each arc of the network with the fire spread time along that arc. Fire travel time through the network was calculated using the Dijkstra’s shortest path algorithm. Using a more accurate input data system, which was able to evaluate the maximum wind speed in 16 directions, the model was later built into an interface model called WUIVAC (Dennison, Cova, & Mortiz, 2007).

Taking a similar approach to Cova et al. (2005), Stepanov and Smith (2012) utilized Dijkstra’s algorithm in their work to predict the wildfire propagation in a heterogeneous landscape. They incorporated a two-pass algorithm based on the Dijkstra’s shortest path algorithm to estimate the minimum travel time paths from ignition points, to specific points of interest. To be able to do this, they designed a methodology based on Delaunay triangulations to represent the landscape with a planar graph. The ellipse model was then applied to calculate the fire spread time along the arcs of the network. The results produced by their model were shown to be similar to the ones produced by FARSITE. FARSITE is considered the most accurate fire simulator by most of the researchers (Papadopoulos & Pavlidou, 2011), although it is not the only popular fire propagation simulator.
In an attempt to make practical use of fire spread models, different simulators and software tools were developed at different times. Fire Area Simulator (FARSITE), is a fire growth simulation modeling system that uses spatial information on topography, fuels, weather, and wind (Mark Arnold Finney, 2004). Existing models for surface fire, crown fire, spotting, post-frontal combustion, and fire acceleration are all incorporated in FARSITE fire growth model. FARSITE is widely used by the National Park Service, U.S. Forest Service, and other federal and state land management agencies for the purpose of simulating the spread of wildfires. FlamMap, developed by Mark A Finney (2006), is another simulator that utilizes the same set of inputs as FARSITE. FlamMap has the ability to map necessary inputs to a landscape level using geographic information systems, extending the utility of models like Rothermel’s, Anderson’s and Alibini’s. FlamMap takes deterministic inputs and was designed to examine “spatial variability” in fire behavior, utilizing a minimum travel time (MTT) algorithm in simulating the fire growth.

Another popular simulator developed by Lopes, Cruz, and Viegas (2002) is FireStation, which is a software system for the simulation of fire spread over complex landscapes. FireStation uses a semi-empirical model to calculate the fire rate of spread that takes slope, fuel, and wind speed and direction as inputs. Depending on the wind speed, it uses either a single or a double ellipse fire model.

FireMaster, a spatial system architecture based on the wavelet model, was developed by Fernandes et al. (2002). FireMaster allows users to simulate bush fire over a distributed network via a map data provided through a World Wide
Web server. This feature provides a framework to reliably broadcast information in case of emergency. FireMaster takes several data layers as input such as vegetation, slope, aspect and fuel load, and provides fire perimeters for different time steps. Other wildfire simulators and software tools that are worth mentioning are Prometheus, FIRE!, DYNAFIRE and PYROCAT. An interested reader may refer to Papadopoulos and Pavlidou (2011) for a comparative review of twenty three different popular simulators, all compared with FARSITE.

2.3 The shortest path problem

The Shortest Path problem (Hargrove et al.) is one of the best-known combinatorial optimization problems and has been extensively studied in the literature. In graph theory, the shortest path problem is defined as the problem of finding a path between two vertices in a graph in a way that the sum of the weights (costs) of edges comprising the path is minimized. The concepts of edge, arc, and link are used interchangeably throughout this review. Weight can be distance, time, cost or any other measuring criterion depending on the nature of the problem involved. The Shortest Path problem has applications in many different areas including operations research, robotics, transportation and communications.

In 1956, Dutch computer scientist, Edsger Dijkstra, introduced Dijkstra's Algorithm, a graph search algorithm that solves the single-source shortest path problem for a graph with non-negative edge weights. The single-source shortest path is a variation of the shortest path problem in which finding the shortest path from the source vertex (node) to all other vertices in the graph is considered.
Since the introduction of Dijkstra’s, there has been a considerable number of algorithms developed to solve the SP problem. Most of them are developed using the principals of Dijkstra’s Algorithm, but usually with a different objective and/or improved run-time complexity. To mention a few (Cherkassky, Goldberg, & Radzik, 1996):

- Bellman–Ford Algorithm solves the single-source problem when edge weights may be negative.
- Tarjan’s Algorithm solves the single-source problem with non-negative edge weights
- A* search Algorithm solves for single-pair shortest path using a heuristic method to speed up the search.

The original shortest path problem is a deterministic problem in which all the edge weights are known in advance and their values are certain. However, for most of the shortest path real-life applications, the deterministic version is not a fitting enough model as it does not consider the various sources of uncertainties involved. An example is the problem of finding the fastest route between two points in a road network where the traffic makes the travel times unpredictable.

To better model uncertainty, one way is to use random variables to represent an uncertain phenomenon. A random variable is capable of grasping the possible states of any uncertain factor, and is a more realistic way of dealing with cases such as road networks. Using this approach, we have a shortest path problem in which the edge weights are stochastic. This problem is called in general the Stochastic Shortest Path problem (SSP).
In contrast to the deterministic version, it is less clear what a stochastic shortest path is as the definition of the “shortest path” we have for the deterministic problem does not apply here. In the deterministic SP, the objective is to find the path with minimum total cost, which can be time or distance. This objective is not meaningful in a stochastic network and we need to define other objective types. For example, the objective can be the shortest path on average, the path minimizing a combination of mean and variance, or the most probable shortest path. As a result, a lot of problem variants have appeared in the literature, each of them focusing on a different objective and with a different formulation.

We will take a look at some of these problem variants and the existing solution approaches. From time-dependency point of view, there are two main categories of the shortest path problem: time-dependent and time-invariant. As the name implies, the concept of time-dependency is only applicable to those networks in which the edge weights are a function of time. Four distinct categories can be defined based on the problem being deterministic/stochastic, and time-dependent/time-invariant (Table 1).

<table>
<thead>
<tr>
<th>The shortest path problem</th>
<th>Deterministic</th>
<th>Stochastic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time-invariant</td>
<td>1. Shortest Path problem (SP)</td>
<td>2. Stochastic SP</td>
</tr>
<tr>
<td>Time-dependent</td>
<td>3. Time-dependent SP</td>
<td>4. Stochastic time-dependent SP</td>
</tr>
</tbody>
</table>

Our approach in this research falls in the second category.

Introducing uncertainty into a network makes it more realistic and at the same time more challenging to model. Optimal routing in stochastic time-dependent
(STD) networks was first addressed by Hall (1986), who considered the minimization of expected travel time in road networks. Hall presented two different ways of formulating the problem. One way is to find a path (route) based on the data in hand, before the actual travel begins, which is often referred to as a-priori route choice. Another way is to devise a time-adaptive strategy for an optimal route choice. In the latter case, instead of determining the entire path, there is an adaptive decision rule that identifies the next immediate edge to take at each node. This strategy is referred to as time-adaptive routing.

Although it may seem otherwise at first, the time-adaptive problem is computationally more tractable. In a discrete STD network, an optimal strategy can be found in linear time (E. Miller-Hooks, 2001). Contrary to the time-adaptive methodology, finding an a-priori route choice is strongly NP-hard (Orda & Rom, 1991). Even though the a-priori route choice is computationally harder, it is the only option in some cases. The case of wildfire propagation is one of them. Fire is not an intelligent being and does not have the ability to decide and adapt to the best route. That leaves the a-priori approach as the only option in modeling wildfire propagation. The analogy between the fire traversal path and the shortest path problem is described later.

One highly desirable objective in a network with stochastic edge weights is finding the distribution of the minimum total weight between two specific nodes, usually a source and a destination. For simplicity’s sake, we call this the distribution of the shortest path. This objective gives a powerful representation of a path in the network and different metrics can be extracted from the distribution such as mean, variance or a confidence interval. This objective is
better expressed with a mathematical formulation. Consider the graph network 
\( G(V, E) \) with \( V \) as the set of nodes and \( e_{ij} \in E \) as the set of edges. Also consider 
\( T_{ij} \) as the random variable representing the weight of edge \( e_{ij} \). \( \text{Pred}(i) \) is defined 
as the set of nodes predecessor to node \( v_i \). Similarly, \( \text{Succ}(i) \) is defined as the set 
of successor nodes. Consider \( s \) and \( t \) to be the indices of source and destination 
nodes, \( v_s \) and \( v_t \), respectively. Also consider \( P \) as the set of all the paths from \( v_s \) 
to \( v_t \), \( P = \{P_1, ..., P_n\} \). \( x^k_{ij} \) is a binary indicator variable that has the value of 1 if \( e_{ij} \) 
is on path \( P_k \), and 0 otherwise. The random variable \( P_{\text{min}} \) has our desired 
distribution and is defined as:

\[
P_{\text{min}} = \min_k \sum_{e_{ij} \in E} T_{ij} x^k_{ij}
\]

\[
\sum_{j \in \text{Succ}(v_i)} x^k_{ij} - \sum_{j \in \text{Pred}(v_i)} x^k_{ji} = \begin{cases} 
1, & \text{if } i = s \\
-1, & \text{if } i = t \\
0, & \text{otherwise}
\end{cases}
\]

\( x^k_{ij} \in \{0, 1\} \)

Frank (1969) was first to consider the problem of finding the shortest path 
distribution in a stochastic framework. His approach was based on an integration 
method that requires first enumerating and then finding the distribution of all 
the possible paths between a source and a destination pair. Theoretically, one 
could obtain the exact distribution of the shortest path by performing the 
integration, but no practical implementation was presented.

V. G. Kulkarni (1986), developed an exact method for the same objective in 
directed networks. He assumed the arc lengths to be independent, exponentially
distributed random variables. The problem was modeled using a continuous-
time Markov chain with a single absorbing state in a way that the time until
absorption represents the length of the shortest path. In a later paper, Corea and
Kulkarni (1993) generalized their method to make it applicable to networks with
integer-valued random variable edge lengths. Although the generalized method
covers more edge length types, its implementation has an extensive computational cost.

Again, with the same objective, Adlakha (1986) tried a Monte-Carlo simulation
approach. Similar to the one of Frank’s, multivariate integrals were utilized to
find the probability distribution of the shortest path and Monte-Carlo was
employed to evaluate the integrals. In order to improve the Monte-Carlo’s
performance, he combined the concept of “unique” arcs (Burt & Garman, 1971)
and “uniformly directed cutset” (Sigal, Pritsker, & Solberg, 1980) to exploit
special properties of the arcs.

Unlike Kulkarni’s, in most of the studies in the literature, finding the probability
distribution of the shortest path is not the main objective. High computational
cost and modeling difficulties seem to be the main reasons for this matter.
Rather, a specific objective function is defined and a solution approach is given
to optimize that particular function. Murthy and Sarkar (1996) considered the
objective of maximizing an expected utility function that is concave and
quadratic. They utilized a label-setting algorithm to identify the Pareto optimal
paths. A relaxation-based pruning technique was then introduced to improve the
efficiency of the label-setting algorithm. The pruning technique was intended to
eliminate unnecessary Pareto optimal paths.
Minimizing the expected travel time of the shortest path is another common objective studied in the literature. Finding the optimal solution in this problem is also challenging, especially in time-dependent networks, as demonstrated with an example by Hall (1986). Looking for the minimum expected shortest path in a time-dependent network, Fu and Rilett (1998) identified a set of relationships between the mean and variance of the travel times of a given path, and the mean and variance of its comprising link travel times. They presented a model based on a continuous-time stochastic process, and introduced a heuristic solution based on the $k$-shortest path algorithm.

Looking at the same objective, E. Miller-Hooks (2001) proposed a two-step solution methodology, which is applicable in a transportation network. In the first step, a-priori least expected time paths from all origins to a single destination were identified. The second step was intended to find a lower bound on the expected times of the a-priori paths discovered in step one.

In a-priori optimization, one strategy is to find the Pareto optimal paths based on the desired cost function utilizing a label-correcting algorithm. In order to find the Pareto optimal paths, path comparison is performed based on a “dominance criterion”. The efficiency and accuracy of these algorithms depend on how well the dominance criteria are defined.

E. Miller-Hooks and Mahmassani (2003) introduced measures for comparing path travel times for both a-priori optimization and time-adaptive choices, in a stochastic time-dependent network. They came up with two dominance conditions, namely pairwise dominance and group dominance to find the Pareto optimal solutions. The expected dominance was another type of dominance introduced by Hutson and Shier (2009). They focused on the problem of finding
the path in a stochastic network that has the minimum combination of the mean and variance of its length.

With a similar objective, Shahabi, Unnikrishnan, and Boyles (2013) studied the mean-standard deviation shortest path problem. In their formulation, path length was defined as a weighted sum of the expected value and the standard deviation of the weight of comprising edges. Accounting for the correlation between edge weights was the main feature of their model. They developed an efficient “outer approximation algorithm” to solve the problem, which demonstrated to be substantially faster than standard algorithms. In a similar approach, Cheng, Lisser, and Letournel (2013) considered the edge weights to be deterministic but having an associated random delay. The delay was represented using its mean and standard deviation with the objective of minimizing the expected path delay penalty.

In another approach looking for the least expected path in a time-dependent network, Chen, Lam, Li, Sumalee, and Yan (2013) proposed a multi-criteria A* algorithm. In their work, they used a discrete approximation of the edges’ travel time distributions. With the same objective, but using a similar approach to Kulkarni’s, Lücking and Stadje (2013) solved the stochastic shortest path problem using Markov Chains with countable state space. They modeled the problem as a negative dynamic programming model to find the optimal policy. A simplified lattice space was used to implement their model.

Ji (2005) studied three different objectives namely “Expected Shortest Path”, “α-shortest Path” and “Most Shortest Path”. “Expected Shortest Path” or the least expected path is the path with the least expected time. “Most Shortest Path” is defined as the path having the highest probability of reaching the destination in a
predetermined time or cost. Considering node $v_s$ as the source and node $v_t$ as the destination in a directed acyclic network, $P$ is the “$\alpha$-shortest path” if (Ji, 2005).

$$\min\{l|\Pr[L(P) \leq l] \geq \alpha\} \leq \min\{l|\Pr[L(P_k) \leq l] \geq \alpha\}, \quad (2)$$

for any path $P_k$ from $v_s$ to $v_t$, where $L(P)$ is the length of path $P$.

In their research, a hybrid intelligent algorithm integrating stochastic simulation and genetic algorithm was developed to solve the “Most Shortest Path” and the “$\alpha$-shortest Path” problems. Later, Gao (2011) studied the relationship between the two problems and proved that there exists an equivalence relation between the “$\alpha$-shortest Path” of an uncertain network, and the shortest path of a corresponding deterministic network. Proving this, he was able to find a solution to the “$\alpha$-shortest Path” problem using an algorithm based on Dijkstra’s. Liu (2010) proved that the “Most Shortest Path” problem is in fact the dual of the “$\alpha$-shortest Path” problem and finding a solution for one, gives the solution to the other.

A-priori reliable shortest path, a concept similar to “$\alpha$-shortest Path” was studied by Nie and Wu (2009). The study was aimed to determine the latest possible departure time and its associated route to achieve a given probability of arriving at the destination. Nie and Wu solved the problem using an exact label-correcting algorithm, which was developed based on the one by E. D. Miller-Hooks (1997).

In this research, one of our main objectives is to find the distribution of the shortest travel-time path from a specified source to a desired destination. As discussed before, finding the exact distribution of the shortest path is an extremely hard goal to achieve, unless all paths from the origin to the destination
are independent (for example, non-overlapping edges) in which case the
distribution can be derived trivially using order statistics. To the best of our
knowledge, there exist no algorithm that is able to find the exact solution in a
timely manner, even for a mid-sized network. Therefore, approximation methods
are required for this purpose. One way to approximate any distribution is to
utilize Monte Carlo simulation by generating sample outcomes of the
distribution. Sigal et al. (1980) used a Monte Carlo simulation approach to
calculate “path optimality index” in a stochastic network. “Path optimality
index” was introduced as an evaluation criterion for paths, defined as the
probability of one path being the shortest among all the other paths in the
network. Their method was intended mainly to estimate the project duration in
PERT analysis, but also applicable for derivation of the minimum time
distribution in a directed acyclic network. V. Kulkarni and Provan (1985)
suggested an improved implementation of the Monte Carlo technique presented
by Sigal et al. to estimate the distribution of the shortest paths in stochastic
networks.

Meybodi and Beigy (2003) introduced a Monte Carlo simulation method, based
on distributed learning automata (Adlakha, 1986), which finds the minimum
expected value of a set of random variables to use in the context of the stochastic
shortest path. Using a similar approach but with the objective of measuring the
network robustness, Morohosi (2010) used Monte Carlo simulation to estimate
the shortest path length distribution between pairs of connected vertices. For
their desired goal, it was required to find this distribution between every
connected pair of nodes in the network. They utilized Monte Carlo simulation to
generate sample graphs in order to reduce the time complexity of their algorithm.

Monte Carlo simulation can also be employed to verify the performance of other algorithms. Fonzone and Schmöcker (2014) studied the effect of link travel time correlation on the performance of a heuristic a-priori route selection method. Performance of the heuristic was evaluated using a Monte Carlo simulation approach. ‘Difficulty’ and ‘importance’ were the two performance measures of a selected shortest path.

In this research, Monte Carlo simulation is utilized for two different objectives, both based on Dijkstra’s shortest path algorithm. The first objective is to estimate the distribution of the shortest travel time path, and the second one is to find the fire stochastic contours. We employ Monte Carlo simulation for both distribution generation and algorithm verification purposes.
3. Wildfire and its behavior

3.1 Introduction

Wildfire behavior is mainly affected by fuel characteristics, topography, and weather. Fuel is any flammable material such as trees, grasses, brush and even houses (2014). The greater an area’s fuel load, the more intense the fire becomes. Dry fuel ignites more easily and burns faster than wet fuel. Topography factors such as slope and landscape also influence the fire propagation. Fire spreads faster uphill as the flames tend to go upwards naturally. Among the weather influencers, wind, temperature and humidity are the main fire drivers. Wind increases the fire spread in its direction. While higher temperatures make the fire burn faster, higher humidity and precipitation may slow it down or even extinguish it. These factors also interact with each other. For example, rain or snow, a weather component, increases the fuel moisture. High humidity slows the drying of the fuel, while winds can make the fuel dry faster. A wildfire could reach the spread rate of up to 14 miles per hour, destroying everything in its path (National-Geographic, 2014). The 1991 Oakland Hills Fire in California alone, destroyed 2,900 structures and cost insurers more than USD 3.5 billion. In 2007 nearly a million people were temporarily displaced in southern California in the "October Fire Siege", causing insured losses of more than $1.3 billion (AIR, 2014). According to the National Interagency Fire Center (NIFC), the cost of fighting U.S. wildfires totaled to $2 billion in 2012. In addition
to the property damage, there is a substantial threat to human life. In 2013 in Arizona, 19 fire fighters lost their lives in the most fatal forest fire fighting event (Holland, 2013).

Many fire models exist, each trying to predict the fire propagation from a different perspective and/or landscape. In this research, we utilize Rothermel’s fire behavior model, an empirical model, to derive the fire ROS. The National Fire Danger Rating System and the FARSITE (Mark Arnold Finney, 2004) software both use Rothermel’s model in their calculations (Scott & Burgan, 2005). In the following subsection we will review the Rothermel’s model and the parameters it takes as input.

3.2 Rothermel’s fire model and BEHAVE

In 1972, Richard Rothermel, a research scientist at the Northern Forest Fire Laboratory in Missoula, Montana, developed an empirical model to predict both fire ROS and fire intensity. Rothermel’s model is currently being used in the National Fire Danger Rating System. His model considers the fire growth through surface fuels, i.e. fuels that are within 6 feet or less of the ground (Rothermel, 1972). Rothermel’s model considers both fuel and environmental characteristics in its input. Fuel characteristics include fuel loading, fuel particle surface-area-to-volume ratio, fuel depth, fuel particle heat content, fuel particle moisture and mineral content, and the moisture of extinction. Environmental inputs of the model are slope of terrain and mean wind velocity. Fuel characteristics are represented in the form of fuel parameters. Fuel models (Table 2) can facilitate this process. They are defined as a “set of fuel parameters that
describe the inherited characteristics that have been found in certain fuel types in the past" (Rothermel, 1972). Fuel models facilitate the parameter calculations and eliminate the need for repetitive measurements.

Table 2: Anderson's 13 fuel models and their parameters

<table>
<thead>
<tr>
<th>Fuel model</th>
<th>Typical fuel complex</th>
<th>Surface-area-to-volume ratio (ft&lt;sup&gt;2&lt;/sup&gt;/fuel loading (tons/acre))</th>
<th>Fuel bed depth</th>
<th>Moisture of extinction</th>
<th>Characteristic surface area-to-volume ratio</th>
<th>Packing ratio</th>
<th>Packing ratio optimum packing ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>1-h</td>
<td>10-h</td>
<td>100-h</td>
<td>Live</td>
<td>Ft</td>
<td>Percent</td>
</tr>
<tr>
<td>Grass and grass-dominated</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Short grass (1 ft)</td>
<td>3,500</td>
<td>0.74</td>
<td>—</td>
<td>—</td>
<td>1.0</td>
<td>12</td>
</tr>
<tr>
<td>2</td>
<td>Timber (grass and underground)</td>
<td>3,000</td>
<td>2.00</td>
<td>100</td>
<td>1.00</td>
<td>30</td>
<td>0.50</td>
</tr>
<tr>
<td>3</td>
<td>Tall grass (2.5 ft)</td>
<td>1,500</td>
<td>3.01</td>
<td>—</td>
<td>—</td>
<td>2.5</td>
<td>25</td>
</tr>
<tr>
<td>Chaparral and shrub fields</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Chaparral (6 ft)</td>
<td>2,000</td>
<td>5.01</td>
<td>100</td>
<td>4.01</td>
<td>30</td>
<td>2.00</td>
</tr>
<tr>
<td>5</td>
<td>Brush (2 ft)</td>
<td>2,000</td>
<td>1.00</td>
<td>100</td>
<td>0.50</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>6</td>
<td>Dormant brush, hard- wood slash</td>
<td>1,750</td>
<td>1.50</td>
<td>100</td>
<td>2.50</td>
<td>30</td>
<td>2.00</td>
</tr>
<tr>
<td>7</td>
<td>Southern rough</td>
<td>1,750</td>
<td>1.13</td>
<td>100</td>
<td>1.87</td>
<td>30</td>
<td>1.50</td>
</tr>
<tr>
<td>Timber litter</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Closed timber litter</td>
<td>2,000</td>
<td>1.50</td>
<td>100</td>
<td>1.00</td>
<td>30</td>
<td>2.50</td>
</tr>
<tr>
<td>9</td>
<td>Hardwood litter</td>
<td>2,500</td>
<td>2.92</td>
<td>100</td>
<td>0.41</td>
<td>30</td>
<td>0.15</td>
</tr>
<tr>
<td>10</td>
<td>Timber (litter and understory)</td>
<td>2,000</td>
<td>3.01</td>
<td>100</td>
<td>2.00</td>
<td>30</td>
<td>5.01</td>
</tr>
<tr>
<td>Slash</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Light logging slash</td>
<td>1,500</td>
<td>1.50</td>
<td>100</td>
<td>4.51</td>
<td>30</td>
<td>5.51</td>
</tr>
<tr>
<td>12</td>
<td>Medium logging slash</td>
<td>1,500</td>
<td>4.01</td>
<td>100</td>
<td>14.03</td>
<td>30</td>
<td>16.53</td>
</tr>
<tr>
<td>13</td>
<td>Heavy logging slash</td>
<td>1,500</td>
<td>7.01</td>
<td>100</td>
<td>23.04</td>
<td>30</td>
<td>28.05</td>
</tr>
</tbody>
</table>

<sup>1</sup>Heat content = 8,000 Btu/lb for all fuel models.
The environmental factors such as wind, slope and moisture are entered on top of the fuel models. Table 2 shows the Anderson 13 fuel models and their parameters, which are inputs to Rothermel’s model.

Rothermel’s model was extended later in Fire Behavior Prediction and Fuel Modeling System (BEHAVE) (Andrews, 1986). BEHAVE is a fuel modeling system and a fire behavior prediction tool that is composed of a set of interactive and user-friendly computer programs (Andrews, 1986). The implementation of the thirteen Northern Forest Fire Laboratory (NFFL) fuel models was added in BEHAVE. In this research, we use a Java implementation of BEHAVE for fire ROS calculations. Rothermel’s model consists of more than 80 equations for predicting the fire intensity and the fire ROS. We will review the main equations in the following subsection.

**Rothermel’s equations**

The main equation among the Rothermel’s equations is the one that calculates the fire ROS (notations are summarized in Table 3):

\[ R = \frac{I_R \xi (1 + \phi_W + \phi_S)}{\rho_b \varepsilon Q_{ig}} \]  

Where \( R \) is the fire ROS in \( ft/min \). This equation demonstrates that the fire ROS is a ratio between the heat received from a source (numerator) and the heat required for ignition (denominator). The numerator has the unit of heat per unit area per time and the denominator has the unit of heat per unit mass which results in the unit of distance per time for the \( R \). Every variable that appears in
(3), is in turn calculated using another set of equations. All the calculations are presented in detail in the Rothermel’s original document (Rothermel, 1972).

Table 3: Important notation used in Rothermel’s equations

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Q_{ig}$</td>
<td>The heat of pre-ignition is the energy per unit mass required for ignition</td>
<td>Btu / lb</td>
</tr>
<tr>
<td>$\varepsilon$</td>
<td>Known as the effective heating number, is a number representing the fuel size, which is near one for fine fuels, and decreases by increasing fuel size</td>
<td>Dimensionless</td>
</tr>
<tr>
<td>$\rho_b$</td>
<td>Bulk density, which is the mass of available fuel per volume unit</td>
<td>lb/ft$^3$</td>
</tr>
<tr>
<td>$I_R$</td>
<td>Reaction intensity, which is the heat release rate per unit area of the fire front</td>
<td>Btu / ft$^2$ min</td>
</tr>
<tr>
<td>$\xi$</td>
<td>Ratio of the propagating flux to the reaction intensity. Propagating flux is the numerator of the $R$ equation.</td>
<td>Dimensionless</td>
</tr>
<tr>
<td>$\beta$</td>
<td>Packing ratio, a value quantifying the compactness of fuel bed</td>
<td>Dimensionless</td>
</tr>
<tr>
<td>$\beta_{op}$</td>
<td>The optimum $\beta$</td>
<td></td>
</tr>
<tr>
<td>$\sigma$</td>
<td>Particle surface-area-to-volume ratio</td>
<td>ft$^{-1}$</td>
</tr>
<tr>
<td>$B, C, E$</td>
<td>Variables calculated using a function of $\sigma$</td>
<td></td>
</tr>
<tr>
<td>$U$</td>
<td>Wind velocity at midflame height</td>
<td>ft/min</td>
</tr>
<tr>
<td>$\tan(\phi)$</td>
<td>Slope, vertical rise/horizontal distance</td>
<td>Dimensionless</td>
</tr>
</tbody>
</table>

The propagating flux is composed of the horizontal flux and the vertical flux. Schematic of propagation flux, illustrating the effects of wind and slope, are shown in Figures Figure 4, Figure 5, and Figure 6. The figures show that the vertical flux in a propagating fire is signified by either increasing the slope or increasing the wind speed.
\( \phi_S \) and \( \phi_W \) are the slope and wind factors respectively, which are applied to adjust the propagating flux. They are functions of fuel, wind and slope parameters. Their equations are as follows:
Wind coefficient:

\[
\phi_W = C U^B \left( \frac{\beta}{\beta_{op}} \right)^{-E}
\]  

(4)

Slope factor:

\[
\phi_S = 5.272 \beta^{-0.3} (\tan \phi)^2
\]  

(5)

3.3 Wind distribution

Wind changes over time, in the order of hours, minutes, or even seconds. To predict the wind speed, it is necessary to know how frequently the wind gusts at different speeds, and therefore to find its speed distribution. We adapt the 2-parameter Weibull distribution to describe and model the wind speed variations as suggested by (Zaharim, Najid, Razali, & Sopian, 2009), (Sarkar, Kumar, & Mitra, 2014), (Lun & Lam, 2000), (Sarkar, Singh, & Mitra, 2011) and (Odo, Offiah, & Ugwuoke, 2012).

As first-order approximation we have assumed the wind direction to be constant during the fire propagation period. The two parameters of the Weibull distribution, \( k_w \) and \( \lambda_w \), are estimated by statistical analysis of the wind speed historical data.

In order to find the wind distribution, the historical wind data of a typical windy day from one of the weather stations close to the study area are analyzed. We chose a windy day for our analysis to present a more realistic weather condition for wildfire propagation. The wind data was acquired from the National Renewable Energy Laboratory (NREL) database which contains wind speed data.
for 10-minute intervals. Various methods are available for curve fitting to real
data. We adapted the method of Rank regression for estimating the wind
distribution parameters.

Weibull distribution

Weibull distribution has various applications in areas such as economics,
inventory control, structural design, reliability analysis, life data analysis and in
our case, in wind speed modeling. The two-parameter (shape, scale) Weibull
distribution has been used for this matter before (Seguro & Lambert, 2000). The
probability distribution function of the two-parameter \((\lambda, k)\) Weibull distribution
is described mathematically as:

\[
f(x, \lambda, k) = \begin{cases} 
\frac{k}{\lambda} \left(\frac{x}{\lambda}\right)^{k-1} e^{-\left(\frac{x}{\lambda}\right)^k} & x \geq 0 \\
0 & x < 0 
\end{cases}
\]  

(6)

With the cumulative distribution function of:

\[
F(x, \lambda, k) = \begin{cases} 
1 - e^{-\left(\frac{x}{\lambda}\right)^k} & x \geq 0 \\
0 & x < 0 
\end{cases}
\]  

(7)

Weibull distribution only takes positive values, which is a legitimate restriction
when used for wind speed modeling. In its formulation, \(k\) and \(\lambda\) are known as
the shape and the scale parameters respectively. When \(k = 1\), the Weibull
distribution reduces to the exponential distribution with scale parameter \(\lambda\).

Figure 7 and Figure 8 illustrate the effect of changing the shape and scale
parameters on the distribution function.

The mean and variance of a Weibull distribution with parameters \(\lambda\) and \(k\) are:
\[ E(X) = \lambda \Gamma \left( 1 + \frac{1}{k} \right) \]  

\[ Var(X) = \lambda^2 \left[ \Gamma \left( 1 + \frac{2}{k} \right) - \left( \Gamma \left( 1 + \frac{1}{k} \right) \right)^2 \right] \]

And the \( n^{th} \) moments are expressed as:

\[ E(X^n) = \lambda^n \Gamma \left( 1 + \frac{n}{k} \right) \]

Where \( \Gamma \) is the gamma function.

The 3-parameter Weibull distribution also known as the translated Weibull distribution has an extra parameter called the location parameter. Its distribution function is described by:

\[ f(x; \lambda, k, \theta) = \begin{cases} 
\frac{k}{\lambda} \left( \frac{x - \theta}{\lambda} \right)^{k-1} e^{-\left( \frac{x - \theta}{\lambda} \right)^k} & x \geq \theta \\
0 & x < \theta 
\end{cases} \]

For wind speed modeling, parameters \( k \) and \( \lambda \) are estimated by analyzing the historical data and finding the best-fitting numbers. Various methods are available for this purpose namely Maximum likelihood method, Rank Regression method, Mean-standard deviation method and Power density method. A comparative study of these methods is presented by Ahmed (2013). We adapt the method of Rank regression for estimating the parameters.
3.4 Fire speed distribution

In this sub-section we derive the fire ROS distribution using Rothermel’s equations and given the wind speed distribution. Wind effect on the fire is itself influenced by terrain and vegetation i.e. fuel type, slope, aspect and humidity. All these influences are quantitatively explained by Rothermel’s equations.

Consider $k_W$ and $\lambda_W$ as the parameters of the wind speed distribution. Also assume that wind is the only stochastic factor affecting the fire ROS, with all the
other factors having fixed values over the period of fire propagation. We prove that ROS follows a 3-parameter Weibull distribution.

**Proposition 1:** Assume \( X \) to be a 2-parameter Weibull random variable with parameters \( k \) and \( \lambda \). Consider the function \( Y = \alpha X^\beta \), with \( \alpha > 0, \beta > 0 \). \( Y \) is a Weibull random variable with parameters \( k^1 = k/\beta \) and \( \lambda^1 = \alpha \lambda^\beta \).

**Proof:** \( Y \) is continuous function of \( X \). Also, with \( \alpha, \beta > 0 \), \( Y \) is a strictly increasing function of \( X \) as \( \frac{dy}{dx} = \frac{\alpha}{\beta} X^{\beta-1} > 0 \). Consider \( f_X \) and \( f_Y \) as the probability distribution functions of \( X \) and \( Y \) respectively. From the probability theory we know that if \( Y \) is an increasing function of random variable \( X \), the distribution function of random variable \( Y \) can be obtained by the equation \( f_Y(y) = f_X(g^{-1}(y)) \frac{dg^{-1}(y)}{dy} \), for \( y \) over the domain of \( Y \), assuming that \( Y \) is differentiable over the domain of \( X \). In our case:

\[
Y = g(X) = \alpha X^\beta \Rightarrow g^{-1}(y) = \left(\frac{y}{\alpha}\right)^{1/\beta} \quad \text{and} \quad \frac{dg^{-1}(y)}{dy} = \frac{1}{\beta} \left(\frac{y}{\alpha}\right)^{1/\beta - 1}.
\]

Also we have \( f_X = \frac{k}{\lambda} \left(\frac{x}{\lambda}\right)^{k-1} e^{-\frac{x}{\lambda}} \). By substituting in the above equations we get:

\[
f_Y(y) = \frac{k}{\lambda} \left(\frac{y}{\alpha}\right)^{1/\beta} e^{-\frac{1}{\beta} \left(\frac{y}{\alpha}\right)^{1/\beta - 1}}
\]

\[
= \frac{k}{\lambda} \left(\frac{y}{\lambda^{k-1} \alpha^{\beta-1}}\right)^{k-1} e^{-\frac{y}{\lambda^{k-1} \alpha^{\beta-1}}} \left(\frac{\lambda^{k-1} \alpha^{\beta-1}}{\lambda^{k-1} \alpha^{\beta-1}}\right)^{1/\beta - 1}
\]

Which gives us:

\[
f_Y(y) = \frac{k}{\alpha \lambda^\beta} \left(\frac{y}{\alpha \lambda^\beta}\right)^{k-1} e^{-\left(\frac{y}{\alpha \lambda^\beta}\right)^k}
\]
By substituting $k' = k/\beta$ and $\lambda' = \alpha\lambda^\beta$ we get:

$$f_Y(y) = \frac{k'}{\lambda'} \left(\frac{y}{\lambda'}\right)^{k'-1} e^{-\left(y/\lambda'\right)^k'}$$

(13)

Which is the distribution function of a 2-parameter Weibull random variable with parameters $k'$ and $\lambda'$ and that concludes the proof.

Now let’s take a look at the ROS in two Rothermel’s equations that state the relationship between ROS and wind speed. We have $R = \frac{iR^\xi(1+\phi_W+\phi_S)}{\rho_b\varepsilon Q_{ig}}$ and $\phi_W = CU^B \left(\frac{\beta}{\beta_{op}}\right)^{-E}$. $U$ is the wind speed which is assumed to have a fixed value in the original equations. In this research, $U$ is no longer a fixed number, but rather a Weibull random variable. Substituting $\phi_W$ into the expression for $R$ we get:

$$R = \frac{iR^\xi \left(1 + CU^B \left(\frac{\beta}{\beta_{op}}\right)^{-E} + \phi_S\right)}{\rho_b\varepsilon Q_{ig}}$$

(14)

And after some algebra:

$$R = \frac{iR^\xi (1 + \phi_S)}{\rho_b\varepsilon Q_{ig}} + \frac{iR^\xi}{\rho_b\varepsilon Q_{ig}} C \left(\frac{\beta}{\beta_{op}}\right)^{-E} U^B$$

(15)

**Proposition 2:** Consider $k_w$ and $\lambda_w$ as the parameters of wind speed ($U$) random variable. Also, assume that wind is the only stochastic factor influencing the fire ROS. $R$ has the 3-parameter Weibull distribution with parameters $k_R = \frac{k_w}{B}$, $\lambda_R = \frac{iR^\xi}{\rho_b\varepsilon Q_{ig}} C \left(\frac{\beta}{\beta_{op}}\right)^{-E} \lambda_w^\beta$, and $\theta_R = \frac{iR^\xi (1+\phi_S)}{\rho_b\varepsilon Q_{ig}}$.

**Proof:** Assume $\delta = \frac{iR^\xi (1+\phi_S)}{\rho_b\varepsilon Q_{ig}}$, $\alpha = \frac{iR^\xi}{\rho_b\varepsilon Q_{ig}} C \left(\frac{\beta}{\beta_{op}}\right)^{-E}$, and $\gamma = B$. That simplifies the equation 15 to $R = \delta + \alpha U^\gamma$. By putting $R' = \alpha U^\beta$ we get $R = \delta + R'$. From
Proposition 1, we can conclude that $R'$ is a Weibull random variable with shape and scale parameters $\frac{k_w}{\gamma}$ and $\alpha \lambda_w^\gamma$ respectively. Assume $k_{R'} = \frac{k_w}{\gamma}$ and $\lambda_{R'} = \alpha \lambda_w^\gamma$ to be the parameters of $R'$. It is trivial to prove that if $f_X(x)$ is the probability distribution function of $X$ for $x \geq 0$, the probability distribution function of $Y = c + X$ is $f_X(y - c)$ for $y \geq c$. Therefore the probability distribution of $R = \delta + R'$ is expressed as:

$$f_R(x) = \frac{k_{R'}}{\lambda_{R'}} \left(\frac{x - \delta}{\lambda_{R'}}\right)^{k_{R'} - 1} e^{-\left(\frac{x - \delta}{\lambda_{R'}}\right)^{k_{R'}}}$$

(16)

With $k_R = k_{R'}$, $\lambda_R = \lambda_{R'}$ and $\theta_R = \delta$ as the shape, scale and location parameters respectively, we have the expression for the 3-parameter Weibull distribution as in equation 10. Using the original notations of Rothermel’s equations we get $k_R = \frac{k_w}{B}$, $\lambda_R = \frac{l_{R\xi}}{\rho b Q_{ig}} C \left(\frac{\beta}{\beta_{op}}\right)^{-E} \lambda_w^B$, and $\theta_R = \frac{l_{R\xi(1+\phi_S)}}{\rho b Q_{ig}}$, which concludes our proof.

### 3.5 Wind distribution from real data

Our study area is located in the Montague city, Massachusetts. We use the wind historical data of a typical day from one of the close weather stations to estimate the wind distribution. The data is acquired from the National Renewable Energy Laboratory (NREL) database. The database contains 10-minute wind speed data for any of selected sites as shown in Table 4. Rank Regression analysis is used for estimating the parameters. Rank regression analysis is a linear regression method that fits the best straight line to a set of points, in an attempt to estimate the parameters of a probability distribution function.
Table 4: Data peek for wind speed

<table>
<thead>
<tr>
<th>Reading Date/Time</th>
<th>Meters per Second</th>
</tr>
</thead>
<tbody>
<tr>
<td>9/2/2014 0:10</td>
<td>6.7</td>
</tr>
<tr>
<td>9/2/2014 0:20</td>
<td>6.8</td>
</tr>
<tr>
<td>9/2/2014 0:30</td>
<td>6.7</td>
</tr>
<tr>
<td>9/2/2014 0:40</td>
<td>6.4</td>
</tr>
<tr>
<td>9/2/2014 0:50</td>
<td>6.2</td>
</tr>
<tr>
<td>9/2/2014 1:00</td>
<td>6.4</td>
</tr>
<tr>
<td>9/2/2014 1:10</td>
<td>5.9</td>
</tr>
<tr>
<td>9/2/2014 1:20</td>
<td>5.8</td>
</tr>
<tr>
<td>9/2/2014 1:30</td>
<td>5.9</td>
</tr>
<tr>
<td>9/2/2014 1:40</td>
<td>6.2</td>
</tr>
<tr>
<td>9/2/2014 1:50</td>
<td>6.3</td>
</tr>
<tr>
<td>9/2/2014 2:00</td>
<td>6</td>
</tr>
<tr>
<td>9/2/2014 2:10</td>
<td>5.9</td>
</tr>
<tr>
<td>9/2/2014 2:20</td>
<td>5.8</td>
</tr>
</tbody>
</table>

In order to find the distribution parameters, the speed data is first binned and normalized to find the wind speed histogram (Figure 9) and then the cumulative histogram is derived (Figure 10).

In order to implement the Rank Regression method, which performs a least square linear regression analysis, a linearization of the cumulative distribution function should be made. The Weibull cumulative function is expressed by

\[ F(x) = 1 - e^{-\left(\frac{x}{\lambda}\right)^k} \]

We can linearize this function by taking the natural logarithm of the two sides, twice:

\[
\ln(-\ln(1 - F(x))) = k \ln(x) - k \ln(\lambda)
\]
This equation has a linear form of \( y' = a x' + b \) where \( y' \) can be equated to \( \ln(-\ln(1 - F(x))) \) and \( x' \) can be equated to \( \ln(x) \). By finding regression coefficients \( a \) and \( b \), the Weibull distribution parameters are calculated as \( k = a \) and \( \lambda = e^{-\frac{b}{a}} \). The linearized data points are plotted in Figure 11 with the least square fitted line. This gives us the slope and intercept numbers of \( a = 1.74 \) and \( b = 2.42 \). Scale and shape parameters of wind Weibull distribution are then derived as \( k = 1.74 \) and \( \lambda = 4 \) respectively. We will use these values in our case study in the subsequent chapters.
Figure 11: Linear regression least square line

Figure 12: Weibull distribution fit to the data histogram
4. Network construction

In this chapter, a methodology to model the fire propagation through a heterogeneous landscape is presented. This methodology is adapted from the work by Stepanov and Smith (2012). In this approach, first a network is constructed, which will represent the fire possible pathways through the landscape. Then, the fire ROS distributions for the network edges are derived.

A heterogeneous landscape is a landscape with varying fire characteristic over different areas. Fire characteristic of an area is identified by features such as humidity, vegetation and terrain. These features all influence the fire ROS. Another important factor influencing the fire ROS is the wind. In this study, wind speed is considered as the only factor with dynamic nature, meaning with varying gusting speed. Other features of a landscape are also prone to change over time. For example, vegetation type may completely change with changing seasons. The reason for selecting wind as the only dynamic factor is that the changes in the wind speed happen with a much faster pace comparing with other factors. In our mode, we consider a period of fire propagation that is not long enough to allow for any perceptible differences in the landscape features. Therefore, we consider them as static.

The original Rothermel’s equations are only applicable to areas with homogenous fire environment i.e. areas with static features. In the previous Chapter we studied the relationship between wind speed and fire ROS distributions. Fire ROS distribution calculation relies on the values of static parameters. Therefore, fire distribution is area-specific and is different for each
homogenous fire environment. That is where the partitioning methodology, which is also known as tessellation comes into play. In simple words, a tessellation of a surface is the tiling of that surface using one or more geometric shapes, in a way that there are no overlaps or gaps.

In order to model the fire propagation, not only do we need a way of calculating the fire ROS, but we also need to specify the possible pathways of the fire. The network, which is built based on the tessellated landscape, is constructed for this purpose. A network is a set of nodes connected by a set of edges and can be directed or undirected. We required a directed network for our representation as the fire ROS is direction dependent.

Stepanov and Smith (2012) used a Triangulated Irregular Network (TIN) for the network. TIN is data model commonly used in Geographic Information Systems (GIS) to represent geography as sets of contiguous, non-overlapping triangles. A grid network is another way of partitioning a geographic area, which is commonly used in fire modeling as reviewed in the literature. One advantage of a TIN compared to a grid network is its considerable smaller size. Another benefit is that a TIN naturally represents areas with higher rate of change (in landscape characteristics) with a denser triangle pattern than areas with a lower rate of change. Grids are far less flexible in coping with variable levels of change (De Wulf, Constales, Stal, & Nuttens, 2012).

Delaunay triangulation is a common approach to construct a TIN from a set of nodes in GIS applications (Tsai, 1993). We adopt the same approach in this research but first we need to construct the network nodes.
4.1 Network nodes

Tessellation was introduced as a way of identifying homogeneous sub-regions on a landscape. Each sub-region has an associated fire environment. A fire environment is a set of different features, namely moisture, vegetation, and terrain, which are expressed by parameter values. These features and therefore the corresponding parameter values are assumed to remain unchanged over the period of fire propagation.

GIS functionality is employed to find the homogenous sub-regions. Each of the abovementioned features is represented by one or more “data layers”. Values of the static factors required for calculating the fire ROS are stored in these data layers. Each layer has its own sub-regions such that the value of a corresponding factor is constant within each sub-region, and changes from one sub-region to another. For example, a data layer for vegetation consists of regions, each with a specific fuel model.

Another feature of the landscape is terrain, which is defined as the horizontal and vertical dimensions of a landscape. Terrain is expressed in terms of elevation, slope and aspect. Aspect is the direction that a slope faces (Bogens, Durfee, Gardner, & Streeper, 2007). The direction that a slope faces can have significant influence on the features of its landscape. This is because of the different angles of the sun in the Northern and Southern Hemispheres. For example, in the Northern Hemisphere, the north side of slopes are often shaded, while in south side they receive more sun. Terrain data is also stored in another data layer.
All the required layers are then *overlaid* to create homogenous regions. This process resembles a big Venn diagram in which a region is a union of different areas, each from a different layer. Figure 13 illustrates an example of the overlaying process for fuel type and elevation layers. Each one of the irregular shaped areas is a region with constant fuel type and height.

After identifying all the regions, a single point is located inside each region, which will replace and represent that region in the network. This process is illustrated in Figure 14 for the same region of Figure 13. The data of all the layers in each region is carried over to the point representing it as its “attributes”. These inside points are a subset of the network nodes. Another set of nodes are added after implementing the Delaunay triangulation step.
4.2 Delaunay triangulation

A triangulation of a set of points $P$ is a “straight-line maximally connected planar graph” with $P$ as its vertices, and with the attribute that its edges only intersect with each other at their endpoints (Lee & Lin, 1986). Delaunay triangulation for a set of points $P$ in a plane is a triangulation that has no point inside the circumcircle of any triangle in the triangulated graph. Minimum angle of all the triangles is maximized in Delaunay triangulations (Mekni, 2010).

Voronoi diagram is the dual to the Delaunay triangulation. In mathematics, a Voronoi diagram is a way of dividing space into a number of regions. Just like Delaunay triangulation, Voronoi diagram requires a set of points called sites to be specified first. A Voronoi diagram of the sites splits the space into different regions such that there is one region for each site. This process is done such that the points in each region are closer to their corresponding site than to any other site (Du & Hwang, 1992). Figure 15 shows the Delaunay triangulation and the Voronoi diagram for the area in Figure 14. Figure 16 illustrates their duality relationship.

In our model, Delaunay triangulation is applied to the nodes inside the sub-regions.
TIN construction can be done in linear time using a “Convex Hull Insertion algorithm” (Tsai, 1993).

Now, we need to calculate the fire ROS on the edges of the constructed network. We rely on the homogenous sub-regions for this calculation as Rothermel’s equations only take static numbers. However, a line connecting two inside nodes will inevitably cross two or more of these sub-regions. We need to partition each line into different edge parts such that each part only overlaps with a single homogenous sub-region. Figure 17 illustrates a situation in which the line connecting the two inside points crosses over 3 different regions. Edges are then replaced by their corresponding edge parts, which will be the new edges.
To complete the network construction additional nodes are added in wherever two or more edge parts meet. This is illustrated in Figure 17. The cross marks represent the new nodes and each part labeled by P represents a new edge. With this arrangement, we can use the Rothermel’s equations to calculate fire ROS given that each edge is now positioned on a single homogenous region.

![Figure 17: Comprising parts of Delaunay triangulated edges](image)

### 4.3 Edge directions

Fire is assumed to traverse along the edges of the network. For most of the edges, the direction that fire would take on that edge cannot be determined in advance. In other words, out of the two head nodes of an edge, it is not known which one would get the flames first. On the other hand, the fire ROS between two adjacent nodes depends on the starting node and the direction that the fire traverses. In order to include this property in the network, each edge between two adjacent nodes is replaced with two directed edges in opposite directions. This arrangement makes it possible to define different fire speeds for the two opposite directions.
4.4 Fire ROS

In order to complete the construction of the network, we need to determine the fire ROS random variable (RV) for all the network edges. Fire ROS random variable is derived by introducing randomness to the fire rate of spread variable that is expressed by one of the Rothermel’s equations. Rothermel’s equations take environmental factors such as fuel and weather characteristics as input. Fuel inputs include fuel loading, fuel particle surface-area-to-volume ratio, fuel depth, fuel particle heat content, fuel particle moisture and mineral content, and the moisture of extinction, which is the fuel moisture content at which a fire will not spread. Wind velocity is one of the most important weather factors, which is why we introduce randomness.

The input values for the fuel including the moisture content are derived from the Anderson fuel models (Anderson et al., 1982) which are available as a data layer for the study area. Six different Anderson fuel models are used in our study area with dead fuel moisture ranging from 12 to 40. Using GIS functionality, we are able to assign the environmental parameters to each edge passing over the sub-region as its attributes. Table 5 shows a peek of the attribute table, which contains the data for each edge.
Table 5: Sample attribute table for the network edges

<table>
<thead>
<tr>
<th>Edge ID</th>
<th>Node from</th>
<th>Node to</th>
<th>Fuel Model (Anderson’s)</th>
<th>Slope Steepness (Degrees)</th>
<th>Aspect (Degrees CW from North)</th>
<th>Edge Forward Direction (Azimuth degrees)</th>
<th>Length (ft.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>7903</td>
<td>5733</td>
<td>279</td>
<td>9</td>
<td>33.6</td>
<td>232.9</td>
<td>209.6</td>
<td>15.0</td>
</tr>
<tr>
<td>7902</td>
<td>38</td>
<td>5733</td>
<td>9</td>
<td>0.5</td>
<td>173.4</td>
<td>209.6</td>
<td>125.5</td>
</tr>
<tr>
<td>7901</td>
<td>5732</td>
<td>611</td>
<td>6</td>
<td>0.4</td>
<td>183.9</td>
<td>291.8</td>
<td>18.7</td>
</tr>
<tr>
<td>7900</td>
<td>5731</td>
<td>5732</td>
<td>8</td>
<td>0.4</td>
<td>183.9</td>
<td>291.8</td>
<td>12.1</td>
</tr>
<tr>
<td>7899</td>
<td>52</td>
<td>5731</td>
<td>8</td>
<td>0.5</td>
<td>173.4</td>
<td>291.8</td>
<td>43.0</td>
</tr>
<tr>
<td>7898</td>
<td>5730</td>
<td>611</td>
<td>6</td>
<td>0.4</td>
<td>183.9</td>
<td>238.7</td>
<td>10.2</td>
</tr>
<tr>
<td>7897</td>
<td>61</td>
<td>5730</td>
<td>8</td>
<td>0.4</td>
<td>183.9</td>
<td>238.7</td>
<td>22.2</td>
</tr>
<tr>
<td>7896</td>
<td>88</td>
<td>5729</td>
<td>1</td>
<td>19.0</td>
<td>283.3</td>
<td>350.3</td>
<td>21.6</td>
</tr>
<tr>
<td>7895</td>
<td>5729</td>
<td>5728</td>
<td>7</td>
<td>19.0</td>
<td>283.3</td>
<td>350.3</td>
<td>0.1</td>
</tr>
<tr>
<td>7894</td>
<td>5728</td>
<td>896</td>
<td>7</td>
<td>0.6</td>
<td>250.7</td>
<td>350.3</td>
<td>86.9</td>
</tr>
</tbody>
</table>

The edge direction is measured by “azimuth”, an angular measurement defined in spherical coordinate system. For that measurement first the vector is projected onto a reference plane. Azimuth is then defined as the angle between the projected vector, and a reference vector on the reference plane. For our application, vector is the vector of a directed edge, and the reference plane is the horizontal plane. Azimuth is the measured in degrees, clockwise from a reference vector pointing towards North, as illustrated in Figure 18.

![Figure 18: Azimuth](image-url)
In this study, a Java implementation of BEHAVE is used as a sub-model to calculate the ROS. There are three major steps to calculate the fire ROS along a specific direction. First, the ROS is obtained without taking into account the wind and terrain influences \(ROS_r\). \(ROS_r\) is calculated using Equation 3, ignoring the wind and slope factors:

\[
ROS_r = \frac{I_R \xi}{\rho_b \epsilon Q_{ig}}.
\]

In the next step, an ellipse is constructed from \(ROS_r\), taking into account the wind and slope factors. The ellipse is characterized by the maximum direction of fire propagation, maximum fire rate of spread \(ROS_{max}\) and eccentricity \(\epsilon\), all calculated based on wind and terrain. The eccentricity of an ellipse is the ratio of the distance between its two focus points to the length of its major axis. It is a number between 0 and 1. An ellipse with eccentricity 0 reduces to a circle and takes a more elongated shape as eccentricity approaches 1. Wind speed has major impact on eccentricity. This is illustrated in Figure 19.

![Figure 19: Effect of wind on eccentricity](image)

In the third step ROS is calculated from the ellipse along the desired direction. This calculation is based on the following formula (Catchpole, De Mestre, & Gill, 1982):
\[ ROS = ROS_{\text{max}} \frac{1 - \epsilon}{1 - \epsilon \cos(\theta)} \]  

(17)

Where \( \theta \) is the angle between the maximum direction and the desired direction.

In order to use this formula in our network, we need to calculate the fire ROS separately for each edge of the network as the fire environments are different for different edges. On each edge, one arbitrary point is selected as the basis for calculations. The fire ellipse is obtained for the selected point based on the attributes of its corresponding edge. At the end, fire ROS is calculated along the edge using equation 17, which is the projected fire speed. This process is illustrated in Figure 20.

The ellipse and the fire ROS in Figure 20 are obtained based on a fixed wind speed. By introducing variability into the wind speed, fire ROS will become a random variable, and instead of one single ellipse, there will be a continuum of infinite number of possible ellipses, each with a different eccentricity. Figure 21 shows an example of some of the possible shapes an ellipse may take, given the wind speed distribution. The fire ROS along an edge is calculated based on the directionless fire ROS, which is calculated from Rothermel’s. This will yield
different projected fire speeds, depending on the different possible wind speeds, as shown in Figure 21.

![Figure 21: Possible projected fire speed](image)

### 4.5 The study area and the real-sized network

We selected the Montague Plains Wildlife Management Area (MPWMA) in West-Central Massachusetts (Figure 23). The reason for this selection is the availability of fire data, which was first provided by Duveneck (2005) and later studied by Stepanov and Smith (2012). The pink area in Figure 22 is the Franklin County and the red section is Montague city.

The Montague Plains Wildlife Management Area (MPWMA) in West-Central Massachusetts was selected as the study area for this research due to the availability of fire data, which was first provided by Duveneck (2005), and later studied by Stepanov and Smith (2012). The MPWMA is owned and managed by the Massachusetts Department of Conservation and Recreation (DCR) Division of Fisheries and Wildlife (DFW) in cooperation with Northeast Utilities (NU). The primary purposes of the site are to preserve and protect an outstanding
example of pine-scrub oak barrens, which occur throughout the Northeast from New Jersey to Maine. The barrens are characterized by excessively drained soils and by several plant species which are highly flammable. The site also provides an area for wildfire viewing and scientific research.

Figure 22: Montague city in Franklin County, MA (Wikipedia)

Figure 23: Montague Wildfire Management Area (Google, 2015)

The resulting network for the study area after overlaying the data layers and triangulating the nodes is presented in Figure 24. The nodes of the network are either inside or on the boundaries of sub-regions. This way it is guaranteed that
each edge only passes through one sub-region. The resulting network for the study area consists of 5733 Nodes and 7049 edges.
Figure 24: Network representation of MPWMA
4.5.1 Why shortest path?

The reason for employing the Dijkstra’s shortest path algorithm in this research is that Dijkstra’s follows the same pattern as the fire propagation in the network. This is better explained in the Table 6

<table>
<thead>
<tr>
<th>Dijkstra</th>
<th>Fire</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial node.</td>
<td>Ignition point.</td>
</tr>
<tr>
<td>Edge length.</td>
<td>Fire traversal time on edge.</td>
</tr>
<tr>
<td>Marking all the nodes other than the initial node as unvisited.</td>
<td>Fire has not reached any other node yet.</td>
</tr>
<tr>
<td>Find/update the tentative distance of the neighbor nodes from the current node.</td>
<td>Calculate/update the expected arrival time of the fire to the neighbor points of the current burning node.</td>
</tr>
<tr>
<td>Mark the node as visited.</td>
<td>The node gets the fire.</td>
</tr>
</tbody>
</table>

One of our objectives is to find the shortest traversal time of the fire from its ignition point to the point of interest. According to Table 6, this can be achieve by implementing Dijkstra’s algorithm on the network with the ignition point as its staring node and with the fire traversal time on each edge as the edge length.
5. **Wildfire front-line stochastic contours**

5.1 **Introduction**

Predicting the fire propagation front-line is a favorable objective among the fire prediction models. The wildfire front-line or perimeter is the dividing line between the burned and unburned parts of a forest. Estimating the fire whereabouts is essential for choosing the mitigation strategy and preparing to evacuate in-danger areas. A precise estimate of the fire perimeter and fire spread rate makes it possible to determine the optimal escape routes.

Fire perimeter modeling has been the subject of study for a considerable time. Models in the literature range from simple models such as the elliptical model (Richards, 1990) to advance fire simulation systems such as FARSITE (Mark Arnold Finney, 2004). However, to the best of our knowledge, there has not been any model that considers the variability of fast changing factors such as wind speed. The major drawback of deterministic models is that the actual fire perimeter almost never matches the predicted perimeter. This effect occurs regardless of the existence of estimation errors. The reason is that in such models, normally an average representation of the possible values is used as the input for fast changing factors. However, in a real fire, it is not the average value that drives the fire, but the actual changing values.

In this Chapter, we present a stochastic fire spread model that is capable of accounting for the changes in the wind speed. The model provides the estimated
location of the fire front-line using probability measures, which are called stochastic contours. The stochastic contours enable the decision maker to determine the reliability of the predicted fire front-line and thus help him in adopting the best action plan.

5.2 Problem statement

Fire propagates in the shape of an ellipse on a homogeneous landscape with constant wind speed (Richards, 1990). However, the landscape is not homogeneous in wildland areas and besides, the wind speed is prone to change and variation. That results in a complex shape for the fire front-line perimeter that changes both from location to location and from time to time. The fire perimeter takes different shapes as it crosses areas with different fire environments. Moreover, the wind speed variations affect the shape of fire front-line over time by changing the fire ROS. In order to predict the wildfire perimeter we need to take into account all these factors.

We are interested in estimating the shape of fire perimeter at time $t$ given the ignition location. However, the expected shape of the fire perimeter alone is not informative enough for prediction as it does not give us any information on the reliability of the estimation.

We introduce the concept of wildfire "stochastic contours", which enables us to estimate the location of the fire perimeter based on different confidence levels. The confidence levels are calculated using probability measures and can help us in determining the reliability of the prediction. Letting the fire ignition time to be
0, at time $t$, the stochastic contour with $\alpha$ confidence is called the "$C_{\alpha}(t)$-contour" and is defined as:

**$C_{\alpha}(t)$-contour (Definition):** $C_{\alpha}(t)$-contour is the smallest contour that encompasses the fire perimeter at time $t$ with $\alpha$ probability.

We are interested in finding the wildfire stochastic contours for different $\alpha$ values. Specifically, we are looking at $\alpha$ values of 0.05, 0.25, 0.5, 0.75 and 0.95. As an example, for $\alpha$ value of 0.95, $C_{0.95}(t)$-contour is interpreted as the imaginary perimeter that encompasses the fire front-line at time $t$ with 0.95 probability. In this research, we utilize Monte-Carlo simulation to derive the stochastic contours. Simulation is implemented by sampling the ROS on each edge and constructing the associated contour at time $t$ around $v_s$ using Depth-Limited Dijkstra, which is our modified version of Dijkstra's shortest path algorithm. This process needs to be repeated many times.

### 5.3 Depth-Limited Dijkstra (DLD)

We introduce the Depth-Limited Dijkstra's algorithm, which we utilize for constructing the fire front-line contours. DLD is developed based on Dijkstra's shortest path algorithm with the objective of finding all the points that are within a certain distance from the source node in a directed network.

Dijkstra's shortest path algorithm is able to find the shortest path from a given source node to all the nodes in a network. Dijkstra's algorithm works using a label-setting procedure in which it starts exploring the graph from the source node and labels all the nodes with the earliest arrival time as the exploration
continues. The arrival time at each node is finalized once all its incoming edges are explored. In our application, we are interested in finding the whereabouts of the fire front-line contour at a specific time, which in simple words is the furthest that the fire has traveled from its ignition point until a certain time. On the network, this translates to all the points that are equally distant from the ignition point by a distance of \( t \) units, which in our case is evaluated based on the corresponding fire traversal time.

DLD is able to find a subset of these points that are located on the network edges by tweaking the exploration step of the Dijkstra's algorithm. Assume the targeted time distance to be \( t \). In the Dijkstra's algorithm, once the arrival time to a node is finalized, the arrival time is compared with \( t \). We add extra steps to modify the algorithm for our purpose. In the modified algorithm, if the finalized time is greater than or equal to \( t \), the node will be taken out from the exploration list and therefore the outgoing edges from the node will no longer be explored.

If the finalized time is exactly equal to \( t \) then the node itself is a point on the contour. However, if the finalized time is greater, we are able to find the point with the exact desired time by extrapolating between the finalized node and its predecessor node on the shortest path leading to the finalized node. We call all the points on the contour the "contour points". It should be noted that this contour is constructed based on fixed generated fire speeds on edges. Later, we integrate this method into a simulation algorithm to find the stochastic contours. Algorithm 1 describes the steps to find the contour points. Following is the definition of notations and functions used in the algorithm steps.
Notations:

\( V \): set of all the nodes in the network
\( E \): set of all the edges in the network
\( t \): contour depth
\( t_{ij} \): time length of edge
\( T(v_i) \): minimum time from the source node \( v_s \) to node \( v_i \)
\( P^C \): set of all the contour points at time \( t \)
\( S(v_i) \): set of all the successor nodes of \( v_i \)
\( Pr(v_i) \): predecessor node of \( v_i \) on the shortest path
\( Q \): set of nodes in the queue to explore
\( v_m \): node with the minimum labeled time in \( Q \)

---

**Algorithm 1 Depth-Limited Dijkstra**

**Input:** \( G(V, E), v_s \) and \( t \)

**Output:** Contour points at time \( t \)

\[
\begin{align*}
Q &\leftarrow v_s \\
T(v_s) &\leftarrow 0 \\
T(v_i) &\leftarrow \infty \text{ for all } i \neq s \\
\text{while } Q \neq \emptyset \text{ do} \\
&\quad \text{find } v_m \text{ where } m = \arg\min_{k[v_k \in Q]} T(v_k) \\
&\quad \text{if } T(v_m) < t : \\
&\quad \quad \text{add all } v_k \in S(v_m) \text{ to } Q \\
&\quad \quad \text{for all } v_k \in S(v_m) \text{ if } T(v_k) > T(v_m) + t_{mk} \\
&\quad \quad \quad T(v_k) = T(v_m) + t_{mk} \\
&\quad \quad \quad Pr(v_k) = v_m \\
&\quad \text{else } P^C = P^C \cup \text{Interpolate}(t, v_m, Pr(v_m)) \\
\text{end while} \\
\text{return } P^C
\end{align*}
\]
The *Interpolate*(t, v_m, Pr(v_m)) function finds the contour point on the edge connecting Pr(v_m) to v_m by interpolating between their coordinates based on the values of T(v_m) and t.

### 5.4 Contour construction

Using the set of contour points with the desired time distance to the source, we are able to construct the complete contour using these points. Contour points are discrete and what we need is a continuous contour. In order to construct the contour, contour points are connected to each other to form a closed shape using "radial sweeping". A north pointing radial line starting from the source node sweeps the contour points clockwise and connects the nodes on its way. This procedure is illustrated in Figure 25.

![Figure 25: Radial sweeping to construct the contours from contour points](image-url)
5.5 Monte-Carlo implementation

$C_\alpha(t)$-contours are constructed using Monte-Carlo simulation of the introduced DLD algorithm. At each iteration of the simulation, edge travel times are sampled according to their corresponding fire traversal time distribution. DLD is then implemented on the network with the depth limit of $t$. Then, the contour for the derived points is constructed using the procedure described in the previous sub-section. This process is repeated many times and the resulting closed contour from each run is stored for integration.

In Monte-Carlo simulation, the accuracy of output improves by increasing the number of runs. In order to estimate the required number of runs to achieve a certain level of accuracy we need to relate the accuracy level to the number of replications. However, we are not able to reveal this relationship, as we have no information on the actual minimum time distribution. Therefore, we take an approach in which we dynamically evaluate the relative error. After each run, the relative error is computed and compared with a predetermined threshold level. The accuracy is assessed using a paired mean-standard deviation measure. In other words, the level of relative change in both the mean and standard deviation is measured as the simulation run proceeds and the algorithm comes to a stop as soon as the relative change for both measures falls below the threshold. The threshold is chosen to be 1%. A combination of the mean and standard deviation provides a reliable representation of the distribution and is easy to compute. To make sure that a below-the-threshold error is not occurring due to chance, a stability period $m$ is established, during which the relative error
percentage needs to be under the threshold for the simulation to terminate, where $m$ is specified in terms of the number of runs.

5.6 Contour integration

The output of Monte-Carlo simulation is a set of closed contours, all constructed using the same $t$, but based on different sampled edge fire travel times. In the next step, $C_\alpha(t)$-contours are derived by integrating the stored contours. The contours are integrated along radial directions. From the fire ignition point, a set of radial lines are drawn, starting from North, clockwise. Each line is one radial degree away from its preceding line making a set of 360 lines on a circular space (Figure 26). For better illustration, only a few contours and only 16 out of 360 radial lines are shown.

![Figure 26: Contour integration](image)
For each line, the intersection of all the contours are determined. As a result, we have a set of intersection points on each radial line that is equal to the number of simulation runs. On each line, the intersection points are then sorted in the order of their direct distance to the ignition point. After that, the $\alpha$ percentile of the sorted points for the targeted $\alpha$ values are determined on each line. Considering the $\hat{\alpha}$ values of 0.05, 0.25, 0.5, 0.75 and 0.95 we have the number of five percentile points on each radial line. Percentile points that correspond to the same $\alpha$ value are then connected together starting from North, clockwise. This results in the construction of $C_{\alpha}(t)$-contours. Each contour can be interpreted according to the way it is constructed. For example, $C_{0.25}(t)$ is interpreted as the contour that encompasses the perimeter of the fire at time $t$ with 25% probability. Finding $C_{\alpha}(t)$ contours for different $\alpha$ values enable us to predict the perimeter of the fire and at same time comment on the reliability of the prediction. For example, one can state that the fire perimeter will be inside the boundaries of contour $C_{0.75}(t)$ by time $t$, with the statement being 0.75% reliable.

5.7 Stochastic evacuation contours

In this sub-section, we introduce the concept of "stochastic evacuation contours", which can be constructed using a similar methodology to the fire stochastic contours. When fire hits an area, it is important to notify the people residing in the path of fire in time to prevent casualties. If the fire gets too close, an evacuation order may need to be put in place. Thus, it is important to know whether the point of interest is within the reach of fire in a specific time frame. This is when the evacuation contours come into use. An evacuation contour of
time $t$ is a contour around the point/area of interest specifying the area within $t$ time unit reach of the fire. Hence, the fire will be less than $t$ time units close to the point of interest as soon as it hits any point on the contour. $t$ is determined in a way that allows enough time for safe evacuation/protection of the people or facilities.

In our model of fire propagation, we consider probabilistic measures for the fire spread. This should be reflected in the evacuation contours. The evacuation contours constructed based on probabilistic fire spread rate are called the "stochastic evacuation contours". The procedure to construct the evacuation contours is as follows. First, all the edges on the network are reversed in direction, maintaining the same parameters as before for the fire rate of spread. Then, the stochastic evacuation contours are created by constructing the stochastic fire contours on the modified network, substituting the ignitions point for the point of interest. The constructed contours are defined and interpreted differently than that stochastic contours. Let $C^E_\alpha(t)$ be the evacuation contour at time $t$ with $\alpha$ confidence level. There is $\alpha$ probability that the fire will reach the point of interest in less than $t$ time units as soon as it hits any points on the $C^E_\alpha(t)$ contour for the first time. An evacuation contour that is closer to the point of interest has a higher corresponding $\alpha$ value and vice versa. This is the opposite of what we had for stochastic contours.

5.8 Experimental results

We implement the stochastic contour algorithm on the study area with one of the network nodes as the ignition point, and with $t$ equal to 46 minutes. After
performing the Monte-Carlo simulation and integrating the resulting contours, the stochastic contours are derived for \( \alpha \) values of 0.05, 0.25, 0.50, 0.75 and 0.95.

The results are shown in Figure 27. The contour for \( \alpha = 0.75 \) shows the area where the fire will be contained within with 75 percent probability when 46 minutes have passed from the ignition time. The other four contours are interpreted in a similar way according to their associated \( \alpha \) value. Were there a situation with no wind variability, all the contours would have merged into one single contour. Conversely, a high variability in the wind speed results in the contours being spread out, especially as more time passes from the ignition time.

The algorithm was developed in Java and was implemented on a desktop station powered by an Intel® Core™ 2 Duo processor with 3.00 GHz clock speed and 4 GB RAM memory, running the 64-bit version of Windows 7.
As expected, we observe that the general shape of contours can be explained using the ellipse model (Richards, 1990). According to the model, in a homogeneous fuel environment with constant wind, the fire takes the shape of an ellipse with its long axis oriented in the direction of the wind direction. On an actual wildland area, fuel, terrain and vegetation all vary across the landscape. In addition, factors such as aspect, slope and wind speed affect the fire direction and speed from location to location, and time to time. The shape of the contours created with our model reflects the effect of all the above-mentioned factors.

5.9 Conclusion

Fire prediction models can help in firefighting efforts to control the wildfire spread and therefore in saving lives and property. The degree of accuracy in fire prediction models depends on the model’s applicability to a given situation and the reliability of the model’s output (Cruz et al., 2013). In fire propagation, wind is one of the most influential, yet unpredictable determinants. In this Chapter we presented a model that accounts for the variability of the wind speed by modeling its effect as a random variable, which makes it applicable to most of the real-life wildfire events. We introduced the concept of fire stochastic contours, which enables us to predict the fire front-line whereabouts given specified \( \alpha \) confidence values.

Most of the simulation software tools for predicting the fire propagation only consider the fire growth in a deterministic way. These models are not capable of capturing the variation effect of the fast changing factors such as wind speed. We utilized Monte-Carlo simulation and a successive integration method to
construct the so called stochastic contours given the ignition point and the time passed from the ignition time. The stochastic contours give the fire predictor the ability to determine the confidence of the prediction and thus enables him to make appropriate decisions accordingly. In addition, we introduced the concept of stochastic evacuation contours, which are constructed using a similar methodology. The evacuation contours can be used to make evacuation decisions based on a desired time buffer and risk level. The time of the contour specified the buffer time and the $\alpha$ value specified the risk level. For example, an evacuation decision based on the 0.05 percent evacuation contour is a conservative decision (low risk) as the fire has at most 5 percent chance of reaching the point of interest in the specified time.
6. Distribution of the shortest fire travel time

6.1 Introduction

In the previous Chapter, we introduced the concept of stochastic contours, which were capable of accounting for the variability of the wind speed in predicting the fire front-line contours. Although fire contours are one of the favorable prediction methods in most cases, there are situations in which the fire front-line location is not informative enough. For example, they cannot provide us with a quantitative enough measure of the risk involved for cases in which residential areas or facilities of high sensitivity or worth are located in the path of fire.

A better alternative is to use time rather than location for the predicting. By calculating the time it takes for the fire to get to a community or an area of interest we are able to make an appropriate decision in a timely manner. Depending on the estimated fire arrival time, this decision could be a partial relocation of some facilities, building a fire break, or complete evacuation of the population.

In this Chapter, we present a stochastic model capable of accounting for the unpredictable changes in the wind speed in estimating the fire travel time. A graph network is used to represent the fire possible propagation paths with the objective of finding the distribution of the source-destination fire travel time. The fire travel time distribution, viewed as a histogram, is derived by Monte-Carlo
simulation. Specifically, ROS on each edge is sampled and the shortest path is calculated using the Dijkstra’s algorithm. This process needs to be repeated many times.

In an event of a fire, it is important to predict the propagation time as fast as possible to be able to react in a timely manner. It is especially important to be able to provide new estimates in case of a change in weather conditions. This emphasizes the need for a model that is sufficiently fast to allow enough time for planning a counter-attack on the fire. Dijkstra’s algorithm run time is directly related to the size of its input network. When employed in a Monte-Carlo simulation, any reduction in its execution time reflects on the total simulation run time. We propose a methodology that substantially reduces the network size by eliminating “non-probable” fire paths. This reduction methodology involves using a dominance criterion that is introduced to evaluate the “worthiness” of the paths in a network. Size reduction is performed in two steps. In the first step, “redundant” nodes are eliminated from the network according to the dominance criterion. In the second step, a heuristic that resembles the implementation of the \( k \)-shortest path algorithm is used to further cut down the network size.

In order to evaluate the effectiveness of this methodology, the distribution of the fire estimated travel time from the reduced-size network is compared with the one from the original network using a Q-Q plot. Also, for a quantitative evaluation, the two histograms are compared using the two-sample Kolmogorov–Smirnov non-parametric test, which is used to test whether two one-dimensional empirical probability distributions differ.
6.2 Fire travel time

Consider a network in which edge weights represent the fire rate of spread distributions. Mathematical expressions to derive these distributions from the wind distribution were presented in Chapter 3. We now calculate the mean and variance of the fire traversal time for each edge.

Let $S$ be the fire spread rate on an edge with length $L$. We are interested in finding the mean and variance of the fire traversal time random variable $\frac{L}{S}$. We showed in Chapter 3 that $S = c + X$ follows Weibull distribution. Let the scale and shape parameters of this Weibull random variable be $\lambda_X$ and $k_X$ respectively. We approximate the mean and variance of $f(X) = \frac{L}{S} = \frac{L}{c + X}$ using the second-order Taylor series expansion of $f(X)$ around $\mu_X$, namely (let $\mu_X$ and $\sigma_X^2$ be the mean and variance of $X$):

$$f(X) \approx f(\mu_X) + f'(\mu_X)(X - \mu_X) + \frac{1}{2}f''(\mu_X)(X - \mu_X)^2$$

(18)

Therefore:

$$E(f(X)) \approx E\left(f(\mu_X) + f'(\mu_X)(X - \mu_X) + \frac{1}{2}f''(\mu_X)(X - \mu_X)^2\right)$$

(19)

$$= f(\mu_X) + f'(\mu_X)E(X - \mu_X) + \frac{1}{2}f''(\mu_X)E(X - \mu_X)^2$$

$$= f(\mu_X) + \frac{f''(\mu_X)}{2}\sigma_X^2$$

For the variance we get:
\[ \text{Var}(f(X)) \approx \text{Var}(f(\mu_X) + f'(\mu_X)(X - \mu_X) + \frac{1}{2} f''(\mu_X)(X - \mu_X)^2) = (20) \]

\[ E\left(f(\mu_X) + f'(\mu_X)(X - \mu_X) + \frac{1}{2} f''(\mu_X)(X - \mu_X)^2\right)^2 - \]

\[ E^2\left(f(\mu_X) + f'(\mu_X)(X - \mu_X) + \frac{1}{2} f''(\mu_X)(X - \mu_X)^2\right) = \]

\[ \sigma_X f'^2(\mu_X) - \frac{1}{4} (\sigma_X^2)^2 f''(\mu_X) + E(X - \mu_X)^3 f'(\mu_X)f''(\mu_X) \]

\[ + \frac{1}{4} E(X - \mu_X)^4 f''(\mu_X) \]

Consider the \( n^{th} \) moment and the \( n^{th} \) central moment of \( X \) to be \( \mu'_n \) and \( \mu_n \), respectively. We have:

\[ E(f(X)) \approx f(\mu'_1) + \frac{f''(\mu'_1)}{2} \mu_2 \]

\[ \text{Var}(f(X)) \approx \mu_2 f'^2(\mu'_1) - \frac{1}{4} \mu_2 f''^2(\mu'_1) + \mu_3 f'(\mu'_1)f''(\mu'_1) + \frac{1}{4} \mu_4 f''^2(\mu'_1) \]

\[ \mu_2 = \sigma_X^2 = \mu'_2 - (\mu'_1)^2 \]

\[ \mu_3 = \mu'_3 - 3\mu'_2\mu'_1 + 2(\mu'_1)^3 \]

\[ \mu_4 = \mu'_4 - 4\mu'_3\mu'_1 + 6\mu'_2(\mu'_1)^2 - 3(\mu'_1)^4 \]

For Weibull distribution, these moments are calculated by:

\[ \mu'_n = E(X^n) = (\lambda_X)^n \Gamma(1 + \frac{n}{k_X}) \]

And the first and second derivatives of \( f(x) \) are given as:

\[ f'(x) = -\frac{L}{(c + x)^2} \]

\[ f''(x) = \frac{2L}{(c + x)^3} \]
The mean and variance of \( f(X) \) can be calculated using equations (21) – (28).

### 6.3 Network size reduction

Utilizing Delaunay triangulation to construct the fire network greatly reduces the number of nodes and edges compared to methods such as lattice network or hexagonal network (Stepanov & Smith, 2012). Despite that, a Delaunay triangulated real-world size network could still be of non-manageable size for simulation purposes.

Monte-Carlo simulation is composed of a number of iterations/runs, each implementing a specific algorithm, which in our case, is the Dijkstra’s algorithm. Reducing the implementation time of each simulation run, even by a small percentage, could greatly reduce the overall implementation time of the simulation. The Dijkstra’s algorithm is able to find the shortest distance/time between a source (ignition point) and a destination (point of interest). Time complexity of Dijkstra’s algorithm is directly related to the number of nodes \(|V|\) and the number of edges \(|E|\) in the network - \( O(|E| + V\log(|V|)) \) (Fredman & Tarjan, 1987).

We propose a two-step size-reduction to manage the size of the network without affecting much the distribution of the fire travel time. The two-step size-reduction relies on the concept of “path dominance”, which we will define in sub-section 6.3.1.
In the first reduction step, some of the non-critical nodes are eliminated. This process is described in sub-section 6.3.2. In the second step, the size of the network is further reduced, this time based on a “minimum-regret” heuristic, which is described consequently in sub-section 6.3.3.

6.3.1 Dominance criterion

Let $G(V, E)$ be a graph representation of our network with $V$ being the set of nodes and $E$ the set of edges. Furthermore, let $P$ be the set of all the paths that originate from node $v_s$ and end in node $v_t$: $P = \{P_1, P_2, ..., P_n\}$. Let $T_i$ be the fire travel time random variable associated with path $P_i$. The minimum fire travel time random variable is $T_{min} = \min(T_1, T_2, ..., T_n)$. Of interest is to find the distribution of $T_{min}$. The fire travel time random variables of the edges in the network are assumed to be independent. Therefore, if path $P_i$ consists of many arcs (as in our case), according to the central limit theorem, the travel time distribution of $T_i$ approximately follows a normal distribution.

**Definition (Dominance Criterion):** Let $\mu_i$ and $\sigma_i$ be the mean and standard deviation of random variable $T_i$, respectively. Consider $P_i$ and $P_j$ to be two of the edge-disjoint paths belonging to set $P$ with travel time random variables $T_i$ and $T_j$ such that $\mu_i < \mu_j$. We say that path $P_i$ $\beta$-dominates path $P_j$ (denoted by $P_i \triangleright^\beta P_j$) if $\mu_i + k_\beta \sigma_i < \mu_j - k_\beta \sigma_j$, where $\phi(k_\beta) = 1 - \beta$ and $\phi$ is the cumulative distribution function of the standard normal distribution.

Intuitively, if $P_i$ and $P_j$ are two potential fire paths, this dominance means that the fire will reach the destination node faster when traveling via path $P_i$ than path $P_j$ and thus $P_j$ can be effectively discarded from further consideration.
Theorem 1. Let $T_i$ and $T_j$ be the travel time random variables of edge-disjoint paths $P_i$ and $P_j \in P$, respectively. Also let $T_{\text{min}}^{ij} = \min(T_i, T_j)$. If $P_i \succ^\beta P_j (0 < \beta < 1)$, then $|F_{\text{min}}^{ij}(t) - F_i(t)| \leq \beta, \forall t$ with $F_{\text{min}}^{ij}$ and $F_i$ denoting the cumulative distribution functions of $T_{\text{min}}^{ij}$ and $T_i$, respectively.

Proof. Assume $L_i = \mu_i - k\beta \sigma_i$ and $U_i = \mu_i + k\beta \sigma_i$. Similarly, $L_j = \mu_j - k\beta \sigma_j$ and $U_j = \mu_j + k\beta \sigma_j$ (Figure 28). By our definition $P_i \succ^\beta P_j$ as $\mu_i < \mu_j$. $T_i$ and $T_j$ are independent random variables as $P_i$ and $P_j$ are edge-disjoint. Therefore $F_{\text{min}}^{ij}(t) = 1 - P(T_{\text{min}}^{ij} \geq t) = 1 - P(T_i \geq t)P(T_j \geq t)$. If $t \geq U_i$ then $P(T_i \geq t) \leq \beta$ which means $1 - F_i(t) \leq \beta$ and also $P(T_i \geq t)P(T_j \geq t) \leq \beta \times P(T_j \geq t) \leq \beta$ which means $1 - F_{\text{min}}^{ij}(t) \leq \beta$.

Knowing that $F_{\text{min}}^{ij}(t)$ and $F_i(t)$ are both non-negative we can conclude that $|F_{\text{min}}^{ij}(t) - F_i(t)| \leq \beta$, which proves the theorem for $t \geq U_i$. If $t \leq U_i$, then $t \leq L_j$ and so $F_j(t) \leq \beta$ meaning $1 - P(T_j \geq t) \leq \beta$. Now, $F_{\text{min}}^{ij}(t) - F_i(t) = 1 - P(T_i \geq t)P(T_j \geq t) - (1 - P(T_i \geq t)) = P(T_i \geq t) - P(T_i \geq t)P(T_j \geq t) = P(T_i \geq t)(1 - P(T_j \geq t)) \leq P(T_i \geq t) \times \beta \leq \beta$ which proves the theorem for $t \leq U_i$ and concludes the proof.

Figure 28: Dominance criterion
In light of the above theorem if $P_i \succ^\beta P_j$, $P_j$ can be effectively dropped from the subsequent calculations to obtain the shortest source-destination time $F_{\text{min}}(t)$. In order to do that the shortest expected path is compared with every other source-destination path in the network.

If the two paths being compared share any edges, only the non-shared part is considered for comparison. The estimation error in calculating the $F_{\text{min}}(t)$ CDF described in the theorem is for a single pairwise comparison. The error increases as the number of eliminated paths grows and the overall error can be revealed using simulation. The total estimation error depends on the topology of the network and the relative values of the means and variances of the paths’ travel times. As it is observed later, we can benefit from a specific characteristic of "fire networks" to keep the overall error small.

This method is for the case of two edge-disjoint paths. In a typical network, instead of just two paths, we have the number of all the source-destination paths. One way to take advantage of the dominance criterion in a network like that is to carry out pairwise comparisons i.e. to compare the shortest expected path with every other source-destination path in the network.

Let $P_{\text{min}}$ be the shortest source-destination expected path. $P_{\text{min}}$ is checked with each of the other source-destination paths, $P_i \in P$, for the dominance status. The non-shared edges of $P_i$ get eliminated if $P_i$ is dominated by $P_{\text{min}}$, and else remain in the network. With this process we aim to attain a smaller sized network that can be used to derive an “accurate enough” approximation of the minimum source-destination travel time distribution.
It should be noted that testing the dominance of one path against another without accounting for their dependency is still a valid test, although a “weaker” one. That is, if two paths are compared without any regard to the shared edges, and one is identified to be dominated, the same result would have been concluded if dependency was taken into account. This is a one-way deduction and the opposite is not necessarily true. In other words if two paths are non-disjoint and one is dominated by the other, we may fail to identify the dominance status by treating them as edge-disjoint.

To prove, consider path $i$ and path $j$ in Figure 29. Let $T'_i$ be the addition of all the travel time random variables of the disjoint edges on path $i$ (colored in blue), and similarly $T'_j$ for path $j$ (colored in green). Assume $T^J$ as the addition of all the random variables of joint edges (colored in red). The dominance of $P_j$ is tested against $P_i$ (considering their dependency) by: $T'_i + k \times \sigma_{T'_i} \leq T'_j - k \times \sigma_{T'_j}$. We show that if $E(T'_i + T^J) + k \times \sigma_{T'_i+T^J} \leq E(T'_j + T^J) - k \times \sigma_{T'_j+T^J}$ then $E(T'_i) + k \times \sigma_{T'_i} \leq E(T'_j) - k \times \sigma_{T'_j}$. That is to show that concluding dominancy from the test without any regard to dependency is a valid conclusion.
Figure 29: Dominance criterion for dependent paths

If \( E(T_i' + T_j') + k \times \sigma_{T_i'T_j} \leq E(T_j' + T_i') - k \times \sigma_{T_j'T_i} \)

\[ \Rightarrow E(T_i') + E(T_j') + k \times \sigma_{T_i'T_j} \leq E(T_j') + E(T_i') - k \times \sigma_{T_j'T_i} \]

\[ \Rightarrow k \times (\sigma_{T_j'T_i} + \sigma_{T_i'T_j}) \leq E(T_j') - E(T_i') \]

\[ \sigma_{T_i'T_j} = \sqrt{\sigma_{T_i}^2 + \sigma_{T_j}^2} \geq \sqrt{\sigma_{T_j'}^2} \]

Similarly \( \sigma_{T_i'T_j} \geq \sqrt{\sigma_{T_i'}^2} \)

\[ \Rightarrow \sigma_{T_i'T_j} + \sigma_{T_j'T_i} \geq \sigma_{T_i'} + \sigma_{T_j'} \Rightarrow E(T_j') - E(T_i') \geq k \times (\sigma_{T_i'} + \sigma_{T_j'}) \]

\[ \Rightarrow E(T_i') + k \times \sigma_{T_i'} \leq E(T_j') - k \times \sigma_{T_j'} \]

The estimation error in calculating the \( F_{\min}(t) \) CDF described in the theorem is for a single pairwise comparison. The error increases as the number of eliminated paths grows and the overall error can be revealed using simulation. The total estimation error depends on the topology of the network and the relative values of the means and variances of the paths’ travel times. As it is observed later, we
can benefit from a specific characteristic of “fire networks” to keep the overall error small.

Figure 30: Worst-case dominance error

To elaborate, consider a case where all the paths from $v_s$ to $v_t$ are edge-disjoint. Let $T_{\text{min}}$ be the minimum travel time r.v. from $v_s$ to $v_t$ and $\bar{F}_{\text{min}}(t)$ be its tail distribution. We have $\bar{F}_{\text{min}}(t) = P(T_1 \geq t)P(T_2 \geq t) \ldots P(T_n \geq t)$. Let the lower and upper bounds of $T_{\text{min}}$ to be $L_1$ and $U_1$ (illustrated in Figure 30). Without loss of generality, assume that $P_{m+1}, \ldots, P_n$ are the final dominated paths as a result of pairwise comparisons. Let $T'_{\text{min}}$ be the minimum time r.v. based on only the non-dominated paths. We are approximating $\bar{F}_{\text{min}}(t) = P(T_1 \geq t)P(T_2 \geq t) \ldots P(T_n \geq t)$ with $\bar{F}'_{\text{min}}(t) = P(T_1 \geq t)P(T_2 \geq t) \ldots P(T_m \geq t)$. The goodness of approximation depends on the value of $o_{m+1} = P(T_{m+1} \geq t)P(T_{m+2} \geq t) \ldots P(T_n \geq t)$. The estimation error of $\bar{F}_{\text{min}}(t)$, $|\bar{F}_{\text{min}}(t) - \bar{F}'_{\text{min}}(t)|$, for $t > U_1$, will still be at most $\beta$ using the same reasoning as for a single pair comparison. For $t \leq U_1$, as $t$ gets lower values, $\bar{F}'_{\text{min}}(t)$ grows larger and the effect of $o_{m+1}$ becomes more perceptible.

The mean and variance of the paths’ travel time random variables relative to each other plays an important role in determining the error. As an example, consider one of the worst case scenarios illustrated in Figure 30. In this scenario,
$T_{\text{min}}$, random variable for $P_{\text{min}}$, has almost zero variance (scales are exaggerated in the Figure for better presentation). There are $k'$ other paths with identical distributions $T_O$, all with infinitely large variances. Assume that the lower bounds of these $k'$ paths are all identical ($L_{\text{other}}$), with a slightly greater value than the upper bound of $T_{\text{min}}$. With this condition, all the paths other than $P_{\text{min}}$ are dominated by $P_{\text{min}}$ and $P_{\text{min}}$ becomes the only non-dominated path. The error in estimating $\bar{F}_{\text{min}}(t)$ by $\bar{F}'_{\text{min}}(t)$ at $t = L_1$ is $\bar{F}'_{\text{min}}(L_1) = P(T_1 \geq L_1)$.

Because of the small variance of $T_1$ and the large variance of $T_O$, $P(T_O \geq L_1) \equiv P(T_O \geq U_1) = 1 - \beta$. Therefore:

$$\bar{F}_{\text{min}}(L_1) = P(T_1 \geq L_1)(T_O \geq L_1)^{k'}$$

$$= \bar{F}'_{\text{min}}(L_1)(T_O \geq L_1)^{k'} = \bar{F}'_{\text{min}}(L_1) \times (1 - \beta)^{k'}.$$

The goodness of approximation depends on the values of $\beta$ and $k'$, and the error increases as the value of $k'$ grows. According to the formula an increase in the variance of non-dominated paths and a decrease in the variance of dominated ones lowers the estimation error. Also, the error decreases by increasing the relative number of non-dominated paths. In a real-world size network simulation can reveal the overall error. As it is examined later in section 6.3.2, the relationship between the means and variances among different paths in a “fire network” does not create a situation close to abovementioned scenario.

The drawback of this approach is that it requires enumerating all the possible paths from the source to a destination, which is cumbersome except in small-size networks. Thus we propose a heuristic reduction algorithm that eliminates the need for the complete enumeration and has a two-step mechanism: 1) node
elimination and 2) path enumeration. In the first step, redundant nodes are removed from the network, and in the second step, the network is further pruned by identifying and eliminating the redundant paths.

Although our methodology does not involve an exhaustive pairwise comparison, it is worth describing how pairwise comparison can be utilized for size reduction in a network with non-disjoint paths. Let \( P_1, P_2, \ldots, P_n \) be all the set of all the paths from \( v_s \) to \( v_t \) with corresponding random variables \( T_1, T_2, \ldots, T_n \) such that \( T_1 \) has the minimum expected value among them. For a pairwise comparison, \( P_1 \) is compared to \( P_i \) to check for dominance status for all \( i \in {2, \ldots, n} \). If \( P_i \) is identified as dominated, all the edge on \( P_i \) that are not shared with \( P_1 \) are marked for removal. Although these disjoint edges are not present with \( P_1 \) are marked for removal. Although these disjoint edges are not present in \( P_1 \), they may be present in other paths that share edges with \( P_i \). Let \( P_j \) be one of those paths. In pairwise comparison with \( P_1 \), if \( P_j \) is identified as non-dominated, all of its marked edges are negated. Some of those negated edges may be marked again in a later comparison. This process goes on until the pairwise comparison is complete for all the paths. The network size is then reduced by eliminating all the edges that remain marked.

6.3.2 Network cut-down step 1: node elimination

We define a node to be \textit{unworthy} if its elimination results in a non-significant change in the distribution of the source-destination minimum travel time. To determine the worthiness of the nodes, we take advantage of a specific property that is typical in a fire network, which we refer to as the “SD tardiness” property.
This property can be explained by the relationship between mean and coefficient of variation among different paths.

**Definition (SD tardiness):** Let $v_s$ and $v_t$ be the source and destination nodes in a network with stochastic arcs. Also, let $T_i$ be the travel time random variable of path $P_i$ from $v_s$ to $v_t$. The network consisting of the set of paths from $v_s$ to $v_t$ is said to be Standard Deviation tardy (SD tardy) if $\frac{\mu_i}{\sigma_i}$ (coefficient of variation) is an increasing function of $\mu_i$ for $i \in N$ where $N = \{1, ..., n\}$, meaning that $\mu_i < \mu_j \Rightarrow \frac{\mu_i}{\sigma_i} < \frac{\mu_j}{\sigma_j}$.

“SD tardiness” implies that an increase in expected value from one path to another in the network occurs with a higher rate than an increase in standard deviation. The reason for the existence of this property in the network lies in the way the fire travel time is calculated for the edges and paths. The fire travel time distribution on each edge is inherited from the wind speed distribution, which has fixed distribution parameters throughout the network. As a result, the mean and SD of fire travel time on different edges tend to scale with each other and thus edges with higher mean typically have a higher SD and vice versa. On the other hand, the travel time distribution on a path is derived from the addition of the travel time distributions of its comprising edges. The mean for a path is calculated using a straight sum of its edge means whereas the SD deviation is calculated using a Euclidean norm. As a result, as the number of edges on a path increases, the SD increases with a lower rate than the mean or is *tardy* in catching up with the mean.

In a typical fire network, higher coefficients of variation are associated with higher means rather than explicitly being an increasing function of the mean. In
other words, this property approximately holds true in the network deeming the network reduction algorithm to be a heuristic. We inspected this property on the study network by randomly selecting 300 paths and plotting their coefficients of variation versus their means in increasing order of the mean value. As it is illustrated in Figure 31. $\frac{\mu_i}{\sigma_i}$ is strongly positively correlated with $\mu_i$, expecting a high accuracy in heuristic results. The “SD tardiness” helps us identify and eliminate dominated paths as shown in the following theorem.

![Figure 31: Coefficient of variation versus mean for randomly selected paths](image)
**Theorem 2.** If the "SD tardiness" property holds for paths $P_i$ and $P_j$ with $\mu_j > \mu_i$, the dominance of path $P_j$ can be concluded from the dominance of path $P_i$.

**Proof.** Consider two cases where 1) $\sigma_j \leq \sigma_i$ and 2) $\sigma_j > \sigma_i$. The proof for case 1 is trivial. For case 2, as $\sigma_j > \sigma_i$ we have:

$$\frac{\mu_j}{\sigma_j} > \frac{\mu_i}{\sigma_i} \Rightarrow \frac{\mu_j}{k\sigma_j} > \frac{\mu_i}{k\sigma_i} \Rightarrow \frac{\mu_j - k\sigma_j}{k\sigma_j} > \frac{\mu_i - k\sigma_i}{k\sigma_i}.$$  

As $\sigma_j > \sigma_i$ we can deduce that $\frac{\mu_j - k\sigma_j}{k\sigma_j} > \frac{\mu_i - k\sigma_i}{k\sigma_j}$ or $\mu_j - k\sigma_j > \mu_i - k\sigma_i$. Therefore if $\mu_i - k\sigma_i > \mu_{\text{min}} + k\sigma_{\text{min}}$ then $\mu_j - k\sigma_j > \mu_{\text{min}} + k\sigma_{\text{min}}$, meaning that the dominance of $P_i$ implies the dominance of $P_j$.  

To proceed with node elimination, we perform a "worthiness" test on each node $v_i$ of the network. In this test, the dominance status of the shortest expected loopless path crossing node $v_i$, $P_{\text{min}}^i$ is assessed with regard to the overall shortest expected path, $P_{\text{min}}$. If $P_{\text{min}}^i$ is dominated, according to the SD tardiness property, we can conclude that all the other paths crossing that node are also dominated and therefore node $v_i$ is deemed "unworthy" and may be safely removed from the network.

To increase efficiency, instead of $P_{\text{min}}^i$, we compare $P_{\text{min}}$ with an easily obtained path whose expected length is a lower bound on the length of path $P_{\text{min}}^i$. This path is obtained by concatenating the shortest expected path from source to $v_i$, $P_{s,i}$, with the shortest expected path from $v_i$ to destination, $P_{i,t}$. Concatenation of the two paths, which is denoted by $P_{s,i} \oplus P_{i,t}$, is achieved by connecting the $v_i$ node on $P_{i,s}$ to the $v_i$ node on $P_{i,t}$ using a zero travel time dummy edge. $P_{i,t}$ is derived by implementing the Dijkstra's algorithm on a modified version of the network in which all the edge directions are reversed, starting from the destination node.
Proposition 2: Consider $\mu_i^{\min}$ as the shortest expected travel time of all the loopless paths from source to destination that go through node $v_i$. $\mu_{s,i} + \mu_{i,d}$ has the same value as $\mu_i^{\min}$ if $P_{s,i}$ and $P_{i,d}$ share no edges and provides a lower-bound on the value of $\mu_i^{\min}$ otherwise.

Proof: Any loopless path from source to destination crossing node $v_i$ is composed of two sub-paths. One that connects $v_i$ to the source and one that connects $v_i$ to the destination. The minimum possible travel time of any path connecting $v_i$ to the source (destination) is $\mu_{s,i}$ ($\mu_{i,d}$) which makes $\mu_{s,i} + \mu_{i,d}$ a lower bound on the value of $\mu_i^{\min}$. If $P_{s,i}$ and $P_{i,d}$ share no edges, a path concatenating the two ($P_{s,i} \oplus P_{i,d}$) is itself a loopless path which makes its travel time equal to $\mu_i^{\min}$ i.e. $\mu_{s,i} + \mu_{i,d} = \mu_i^{\min}$. We are now able define the “worthiness” of a node.

Definition: Let $P^i$ be the concatenation of $P_{s,i}$ and $P_{i,d}$ at node $v_i$ ($P^i = P_{s,i} \oplus P_{i,d}$). Node $v_i$ is identified as “unworthy” (redundant) if $\mu^i - k\sigma^i > \mu_{\min} + k\sigma_{\min}$.

With this definition, the test is carried out for all the nodes in the network and a node is eliminated if it is deemed unworthy. This is concluded based on two premises. First, $\mu^i$ is a lower bound value for the expected travel-time of any path from source to destination that crosses node $v_i$. Second, according to proposition 1, $\mu_i - k\sigma_i$ is an increasing function of $\mu_i$. Taking these two premises into account we can conclude that if $\mu^i - k\sigma^i > \mu_{\min} + k\sigma_{\min}$ then $\mu_k - k\sigma_k > \mu_{\min} + k\sigma_{\min}$ for any path $P_k$ that crosses node $v_i$. In other words $\mu^i - k\sigma^i > \mu_{\min} + k\sigma_{\min}$ shows us that all the possible paths from the source to destination that pass through node $v_i$ are dominated by $P_{\min}$ and therefore node $v_i$ may be safely removed from the network.
One might question the credibility of the method because of the existence of paths that share edges with $P_{min}$. This is when the extended definition for dominance criterion comes into play. According to the extended dominance criterion ignoring the dependency of paths weakens the dominance criterion but maintains its credibility. In implementation terms this translates to a weakened discerning ability to identify redundant nodes and therefore a possible larger size output network. The reason for ignoring the shared edges in step 1 is to lower the algorithm complexity considering the number of potential comparisons needed in a real-world size network. Dependency will be accounted for in step 2 as the number of required comparisons becomes much fewer. Algorithm 2 describes the node elimination step.

```
Algorithm 2  Node Elimination Step
Input: $G(V,E)$
Output: $G_N(V,E)$

$P^0 \leftarrow \text{FindShortestPath}(G(V,E), v_s, v_t)$
$S^s \leftarrow \text{ShortestPathsFrom}(G(V,E), v_s)$
$S^t \leftarrow \text{ShortestPathsFrom}(G(V,E^R), v_t)$
for $v_i \in V$ do
    if $P^0 \not\sim^\beta (S^s \oplus S^t)$
        then $V \leftarrow V - v_i$
end for
$G_N(V,E) \leftarrow \text{Trim}(G(V,E))$
return $G_N(V,E)$
```

ShortestPathsFrom(network, node) utilizes Dijkstra's to find the set of all the shortest paths on its input network from its input node to all the nodes. Those paths are all stored in $S^s$ with $S^t$ be the one among them that ends in node $v_i$. $G(V,E^R)$ is the network $G(V,E)$ where all the edge directions are reversed.
Similar to \( S^i, S^t_i \) is the shortest path from \( v_t \) to \( v_i \) in \( G(V, E^R) \), which translates to the shortest path from \( v_i \) to \( v_t \) in \( G(V, E) \). \textit{Trim} eliminates edges in set \( E \) that are not directly connected to any of the nodes in set \( V \).

It should be noted that in the process of reversing, only the edge directions are changed and the fire travel time parameters remain intact. \( P_{s,i} \oplus P_{i,t} \) is the shortest path from \( v_s \) to \( v_t \) crossing \( v_i \) without the restriction of needing to be loopless throughout the path. Therefore \( P_{min}^i \), which is derived with that restriction imposed, has either the same or a worse (higher) expected time than \( P_{s,i} \oplus P_{i,t} \). Thus, the expected length of \( P_{s,i} \oplus P_{i,t} \) is a lower bound on the expected length of \( P_{min}^i \) and the dominance of \( P_{min}^i \) can be concluded from the dominance of the easily obtained \( P_{s,i} \oplus P_{i,t} \). \( P^0 \) is the shortest expected path from \( v_s \) to \( v_t \).

6.3.3 Network cut-down step 2: path enumeration

We further cut down the network size using the following heuristic. We identify a "restrained area" on the network that contains the non-dominated paths and eliminate any nodes/edges that fall out of the boundaries. One way to implement this is to utilize an exact method such as the loopless \( k \)-shortest path algorithm. In the \( k \)-shortest path algorithm, paths are discovered in increasing order of expected length. Each discovered path is checked for dominance against the first discovered path. When the first dominated path is revealed, according to the \( SD \)-tardiness property we can conclude that all the non-explored paths, which have higher means, are also dominated.
One way to implement this is to utilize an exact method such as the loopless \( k \)-shortest path algorithm. In the \( k \)-shortest path algorithm, paths are discovered in increasing order of expected length. Each discovered path is checked for dominance against the first discovered path. When the first dominated path is revealed, according to the \( SD \)-tardiness property we can conclude that all the non-explored paths, which have higher means, are also dominated. Let \( P_{min_k} \) be the \( k^{th} \) discovered path. We are interested in finding \( P_{min_k} | \mu_{min_k} - k\sigma_{min_k} \geq \mu_{min} + k\sigma_{min} \) with the lowest \( k \) value. The first \( k \) paths will then stay in the network and all the other parts of the network will be discarded.

![Source](Image)

![Destination](Image)

**Figure 32: \( k \)-shortest path in a planar triangulated graph**

The rationale behind this process can be explained using proposition 1. Proposition 1 states that if a path is dominated, all the paths with a greater expected length (time) are also dominated. In the \( k \)-shortest path algorithm, paths are discovered in increasing order of expected length. By discovering the
first dominated path we can conclude that all the non-explored paths are dominated.

The limitation of this method is that the number of paths that needs to be enumerated is of an extremely high order, which contradicts the aim of network reduction. Our network is a planar triangulated network where the $i^{th}$-shortest path is most probably only slightly different from the $(i - 1)^{th}$-shortest. This is illustrated in Figure 32. The red path is the shortest expected path and each other color illustrates one deviation that leads to one of the longer shortest paths. Five of these paths are displayed in the Figure, each different from the first shortest path by only two edges. In a real-world size network this effect becomes more appealing. Thus, the number of paths we need to enumerate to find the first dominated path will be extremely high.

To overcome this dilemma, we propose a heuristic that resembles the implementation of the $k$-shortest path but avoids the redundant enumerations by skipping over the non-dominated paths. To better describe this algorithm we introduce the concept of $d$-deviation. Assume that the shortest expected path from $v_s$ to $v_t$ on network $G(V, E)$ is $P_1$. A path from $v_s$ to $v_t$ is called $d$-deviated if its expected travel time is $d$ units larger than the travel time of $P_1$. The algorithm finds the first dominated path by exploring the $d$-deviated paths.

Let $d_{\text{max}}$ be the maximum deviation among all the non-dominated paths. Our aim is to only explore the $d$-deviated non-dominated paths using the heuristic and to use them to identify all of the non-dominated paths, including those that were skipped over. In order to do that, an area that minimally encompasses the non-dominated $d$-deviated paths with $d \leq d_{\text{max}}$, referred to as the "restrained
area”, is constructed. Then, all other parts of the network that fall outside the restrained area are eliminated.

Figure 33: Minimal area with two components

This heuristic takes advantage of the fact that the edges of the network represent the fire traversal paths and thus the fire spread rate is maximum along the effective wind direction and slows down as it deviates from this direction. The restrained area ideally encompasses only the non-dominated paths such that all the $d$-deviated paths with $d > d_{max}$ will have at least one edge outside its boundaries.

It should be noted that the heuristic is capable of handling the cases where the wind ellipse effect is not present although the process of constructing the
“restrained area” from the deviated paths will become more challenging (Figure 33).

**Algorithm 3** Path Enumeration Step

**Input:** $G_N(V,E)$

**Output:** Restraimed area $R$

1. $P^0 \leftarrow \text{FindShortestPath}(G_N(V,E), v_s, v_t)$
2. $P^d \leftarrow P^0$
3. $E^s, M \leftarrow \emptyset$
4. while $(P^0 \ominus E^s) \neq (P^d \ominus E^s)$ do
   5. $M \leftarrow M \cup P^d$
   6. $e_{MRE} \leftarrow \text{FindMinRegEdge}(P^d)$
   7. $E \leftarrow E - e_{MRE}$
   8. $P^d \leftarrow \text{FindShortestPath}(G_N(V,E), v_s, v_t)$
   9. $E^s \leftarrow \text{FindSharedEdges}(P^0, P^d)$
5. end while
6. $R \leftarrow \text{FindRestraimedArea}(G_N(V,E), M)$
7. return $R$

Let $M$ be the set of all explored $d$-deviated paths with $d \leq d_{\text{max}}$. Algorithm 3 describes the steps for enumerating the $d$-deviated paths and identifying the "restrained area".

$\text{FindMinRegEdge}$ finds the "Minimum regret edge" of its input path. "Minimum regret edge", denoted by $e_{MRE}$, is the edge whose removal results in the minimum increase in the length of the re-calculated shortest path. This variable is used to adjust the changes in $d$. $P^d$ is the $d$-deviated path and is initialized by $P^0$. The $\ominus$ sign is the path subtraction sign. $P^0 \ominus E^s$ is obtained by collapsing the $E^s$ edges on $P^0$. 
FindSharedEdges returns the shared edges between its two input paths. The output is stored in $E^s$. During the loop, the algorithm repeatedly identifies the "minimum regret edge" of a recently discovered deviated path and eliminates that edge, then for the next deviated path. This process goes on until a dominated deviated path is discovered for the first time. The value of $d$ for the last non-dominated $d$-deviated path $P^d$ found is $d_{max}$.

FindRestrainedArea($G_N(V,E), M$) finds the sub-network of $G_N(V,E)$ that is on or inside the boundaries of the set of explored paths, $M$. This sub-network, which is the output of the algorithm, is stored in $R$.

There are two main differences between this heuristic and the $k$-shortest path algorithm. First, instead of $k$, the dominance criterion is used as the termination rule. Second, the concept of $d$-deviated path is utilized instead of the $k$-shortest path. The $d$-deviated approach tries to overcome the slow-moving problem of the $k$-shortest path algorithm by taking larger steps and skipping over highly similar paths. At the same time, the heuristic tries to avoid taking “too large” skipping steps, which may lead to a restrained area with almost identical size to the input network. Minimum regret edge calculations are to adjust the skipping steps (changes in $d$).

When the first dominated path is discovered, $D$ is populated with all the explored non-dominated $d$-deviated paths. The minimal area is then constructed using the paths in $D$ using the original network $G(V,E)$ as input. First, all the nodes that belong to the paths in set $D$ are marked on the original network. An arbitrary node outside the minimal area is then selected by visual inspection and the network is explored starting from the selected node. The exploration is done
using the Breath-First-Search algorithm on the condition that no edge may be explored from a marked node. When there are no more reachable edges to explore, all the explored edges are removed from the network, which only leaves the edges of the minimal area on the network. This process is illustrated in Figure 34.

![Diagram of network with nodes, edges, and markings demonstrating the process of finding the minimal area.]

Figure 34: Finding the minimal area

On the Figure, starting from the corner “starting node”, the algorithm marks the explored edges for removal. The directions on the green edges show the direction towards which the algorithm advances. The blue edges are not explored as they are originating from a node belonging to set \( D \) (shown in red). The minimal area composing of all the non-dominated paths remains after removing the marked edges.
It should be noted that a minimal area is not necessarily convex and may even consist of more than one component, with components only connected at the source and destination nodes. This is illustrated in Figure 33. The red edges are the ones belonging to set \( D \). The area specified in light blue is the minimal area. A very steep slope or a barrier in the area between the two compartments may cause the minimal area to appear in a shape like this.

6.4 Shortest time histogram

Our goal is to find the distribution of the fire source-destination travel time. Monte-Carlo simulation approach is selected for this purpose.

6.4.1 Monte-Carlo simulation

Monte Carlo simulation is a class of simulations that relies on repeated random sampling. It is typically utilized when it is impossible or unfeasible to obtain an exact result or a closed form expression. One application is to obtain the distribution of an unknown probability distribution by sampling from the distribution over a repeated number of runs. The results from the runs are analyzed and aggregated to estimate the distribution.

We have a graph \( G(N, V) \) with a source node \( v_s \) and a destination node \( v_t \). Fire is assumed to travel along the edges with a speed that is dependent on each particular edge. The fire speed on each edge is characterized by a Weibull random variable associated with that edge. The fire speed is generated for all the network edges in each simulation run according to the corresponding random variables. The fire traversal time is then calculated from the generated fire speed
and the edge length. Using the fire traversal times, Dijkstra’s algorithm is employed to find the shortest traversal time from \( v_s \) to \( v_t \), and the calculated shortest times from each run gets recorded.

Fire speed random variables are generated using the inverse probability integral transform method. Inverse probability integral transform is a method to generate random numbers of a distribution given its cumulative distribution function. Probability integral transform states that the cumulative distribution function \( F(X) \) of a continuous random variable \( X \) has a uniform distribution on ‘0-1’. Hence, one can draw a random number \( x \) from \( X \) by first generating a random number \( u \) from ‘0-1’ uniform distribution and find the \( x \) such that \( F(x) = u \).

Fire spread rate on each edge follows the 3-parameter Weibull distribution. Assume the Weibull parameters of the random variable \( x \) of one particular edge to be \( \lambda, k \) and \( \theta \). We have \( F(x, \lambda, k, \theta) = 1 - e^{-\left(\frac{x-\theta}{\lambda}\right)^k} \) for \( x > \theta \). By generating the random number \( u \) and putting \( 1 - e^{-\left(\frac{x-\theta}{\lambda}\right)^k} = u \) we get \( x = \theta + \lambda(\ln(1-u))^\frac{1}{k} \). This is the formula that is used to generate sample fire rate of spread for each edge.

At the end of simulation we have the set of all the recorded shortest times, each calculated based on a different sampled network data. The recorded shortest times are then summarized in a histogram.

In Monte Carlo simulation a more accurate result is obtained by increasing the number of runs. The number of runs is thus determined based on the desired accuracy. This number is easily obtainable if there is a clear relationship between the output accuracy and the number of runs. For example, the number of runs is
calculated by \( \left( \frac{Z_{\alpha} \sigma_x}{\text{err}} \right)^2 \) for a case where Monte Carlo simulation is used to estimate the mean of a Normally distributed random variable \( X \) with known variance \( \sigma_X^2 \). Here \( \text{err} \) is the desired accuracy/error.

This approach is not applicable in our problem for two main reasons. First, we are not able to find the mathematical expression as the distribution type is unknown. Second, we are comparing two distributions and not only one parameter such as mean. We take an approach in which the simulation is continued until a stopping criterion is met, which is defined based on a dynamically calculated error.

**Stopping criterion**

By dynamically evaluating the error we no longer rely on the relationship between the accuracy and the number of runs. However, the choice of the metric for error calculation is still an issue. The ideal metric should be able to evaluate the change in the shape of distribution in each run. One potential approach is to find the histogram of all the collected data until each run and compare it with the one until the previous run using a similarity test. This approach requires a significant amount of calculations and is not feasible.

Our solution is to use a metric that is assessed using a paired mean-standard deviation measure. In other words, the level of relative change in both the mean and standard deviation is measured as the simulation run proceeds and the algorithm comes to a stop as soon as the relative change for both measures falls below the threshold. Threshold is chosen to be 1%. A combination of the mean
and standard deviation provides a reliable representation of the distribution and is easy to compute.

The threshold value is expressed in terms of relative change percentage. Let $M_n$ and $S_n$ be the mean and standard deviation of the shortest time data until the $n^{th}$ run. The relative error percentage of these two criteria is defined as $\frac{|M_{n-1} - M_n|}{M_n} \times 100$ and $\frac{|S_{n-1} - S_n|}{S_n} \times 100$. The simulation goes on until the error percentage for both the mean and standard deviation falls below 1%.

To make sure that a below-the-threshold error is not occurring due to chance, a stability period $m$ is established, during which the relative error percentage needs to be under the threshold for the simulation to terminate, and $m$ is specified in terms of the number of runs.

### 6.5 Evaluation: Original vs. reduced-size network

The simulation run time is directly related to the network size. This can be expressed mathematically using the big-O notation. For a network $G(V, E)$, the run time of the Dijkstra’s using the Fibonacci heap implementation of the min-priority queue is $O(|E| + Vlog(|V|))$ with $|V|$ and $|E|$ as the number of nodes and the number of edges in the network respectively (Fredman & Tarjan, 1987). The overall run time of the simulation with $N$ iterations will be $O(N(|E| + Vlog(|V|)))$.

In order to validate the derived empirical distributions, the distributions from the reduced-size network are compared with the ones from the original network.
The ultimate use of the fire travel time distributions will be in the form of cumulative distribution values, which are used to estimate the likelihood of fire arrival time to the point of interest at a specific time and/or estimate the arrival time given a specific probability. Therefore, we calculate the cumulative distributions for this purpose.

**Similarity test**

We carry out a comparison test both visually, using a $Q-Q$ plot, and quantitatively, using the Kolmogorov-Smirnov (KS) two-sample non-parametric test. $Q-Q$ plot visually accentuates the difference between the two distributions being compared. The $Q-Q$ plot will lie on the $y = x$ line if they are exactly alike. An arc or "S" shape in $Q-Q$ plot indicates that one of the distributions is more skewed, or has heavier tails, than the other. The KS two-sample test on the other hand quantifies the distance between the empirical distribution functions of two samples (Massey Jr, 1951) and is used to test whether two one-dimensional probability distributions differ. The null distribution of the test hypothesis is calculated under the assumption that the two samples are drawn from the same distribution. The test is sensitive to the difference in both location and shape. In order to calculate the KS statistic, the range of values from the two histograms is divided into hundred bins and the empirical cumulative distribution value of each histogram is calculated at the starting point of each bin. The value of KS statistic is compared to the critical value of $\alpha=0.05$.

The empirical distribution function $F_n$ for $n$ iid observations $x_i$ is defined as:

$$F_n(x) = \frac{1}{n} \sum_{i=1}^{n} I\{x_i \leq x\}$$
Where $I(.)$ is the indicator function which is equal to 1 if $x_i \leq x$ and 0, otherwise. $F_n(x)$ is basically equal to the fraction of $x_i$’s that are less than or equal to $x$.

The KS statistic for the two-sample test is:

$$T_{n_1,n_2} = \sup_x |F_{n_1}^1 - F_{n_2}^2|$$

Where $F_{n_1}^1$ and $F_{n_2}^2$ are the empirical cumulative distributions of the two samples which in our case are the samples from the original and reduced-size networks. KS statistic tests the following hypothesis:

$$H_0: F_{n_1}^1(x) = F_{n_2}^2(x) \text{ for all } -\infty \leq x \leq \infty$$

$$H_1: \text{Otherwise}$$

The null hypothesis is rejected at the level of $\alpha$ if:

$$T_{n_1,n_2} > c(\alpha) \sqrt{\frac{n_1 + n_2}{n_1 n_2}}$$

The values of $c(\alpha)$ for different $\alpha$ levels are given as:

<table>
<thead>
<tr>
<th>$\alpha$</th>
<th>0.001</th>
<th>0.0050</th>
<th>0.01</th>
<th>0.05</th>
<th>0.10</th>
</tr>
</thead>
<tbody>
<tr>
<td>$c(\alpha)$</td>
<td>1.95</td>
<td>1.73</td>
<td>1.63</td>
<td>1.36</td>
<td>1.22</td>
</tr>
</tbody>
</table>

### 6.6 Experimental results

We demonstrate the value gained from the network size reduction methodology by comparing the size of the original network with the reduced-size network, and also its effect on the simulation run time. To carry out a fair run-time
comparison, we pre-set the number of runs to be 50000 for all experiments. The algorithm was developed in Java and was implemented on a desktop station powered by an Intel® CoreTM 2 Duo processor with 3.00 GHz clock speed and 4 GB RAM memory, running the 64-bit version of Windows 7. For each network/scenario combination, the simulation results are averaged over ten replications to assure the robustness of the output number. It should be noted that the simulation for the original network is only performed for validation purposes and it is not fast enough to use in real-life fire events.

Fire travel time mean and standard deviation needs to be calculated from fire speed distribution on each edge in both directions in order to implement the two-step reduction methodology. Table 7 shows a peek of the table with calculated mean and standard deviation values.

<table>
<thead>
<tr>
<th>Node ID (From)</th>
<th>Node ID (To)</th>
<th>Length (Meter)</th>
<th>Forward Time Mean (Seconds)</th>
<th>Forward Time STD (Seconds)</th>
<th>Backward Time Mean (Seconds)</th>
<th>Backward Time STD (Seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1398</td>
<td>1203</td>
<td>59.6</td>
<td>67.0</td>
<td>24.6</td>
<td>2388.9</td>
<td>661.3</td>
</tr>
<tr>
<td>159</td>
<td>171</td>
<td>10.9</td>
<td>2740.6</td>
<td>1130.4</td>
<td>40.1</td>
<td>10.7</td>
</tr>
<tr>
<td>156</td>
<td>192</td>
<td>64.7</td>
<td>4561.6</td>
<td>1303.3</td>
<td>1496.6</td>
<td>505.2</td>
</tr>
<tr>
<td>1542</td>
<td>1507</td>
<td>5.0</td>
<td>115.1</td>
<td>47.2</td>
<td>2079.0</td>
<td>678.3</td>
</tr>
<tr>
<td>1393</td>
<td>1411</td>
<td>5.2</td>
<td>4002.2</td>
<td>1376.1</td>
<td>214.2</td>
<td>87.8</td>
</tr>
</tbody>
</table>
6.6.1 Network pruning

To illustrate the benefit of the network reduction algorithm we selected two nodes as the source and destination (Figure 35).

The figure is color-coded according to the vegetation type. The edge colors change from red to green in a continuum as the vegetation type changes from a kind that allows for higher fire speed to one that limits the fire speed. Four scenarios with varying wind directions are considered: a) 0, b) 90, c) 180 and d) 270. The direction of the wind is the direction that the wind is blowing to, starting from the North and going clockwise.
The $\beta$ level for the dominance criterion is set to the three sigma level of the normal distribution ($Z_\beta \equiv 3$). The output of the nodes elimination step for the
The number of remaining nodes after the node elimination step depends on the direction of the wind. As expected, we observe that the highest reduction in the number of nodes occurs when the wind speed approximately aligns with the source-destination direction, which can be partially explained using the ellipse model ([Richards, 1990]). According to the ellipse model, in a homogeneous fuel environment with constant wind and weather conditions, the fire takes the shape of a simple ellipse with the long axis of the ellipse oriented in the direction of the wind direction. Figure 37 illustrates the effect of wind direction on fire propagation explained by the ellipse model. The ellipse model shows how the wind direction becomes more closely aligned with the source-destination vector, and how the fire moves toward the destination in a narrower path. We call this the “ellipse effect”.

Figure 37: The effect of wind direction explained using the ellipse model for wind directions: (a) 0, (b) 90, (c) 180 and (d) 270.

four scenarios is illustrated in Figure 36 where the output nodes are marked in red.
Figure 38: Output networks depicting the nodes from node elimination step as well as the generated paths from the path enumeration step for wind directions: (a) 0, (b) 90, (c) 180 and (d) 270.

Figure 38 shows the output networks after the path enumeration step for the four wind scenarios, where the boundaries of the enumerated paths, which are
colored in blue, define the restrained area. The size of the path enumeration step can be compared with the size of the node elimination step.

![Diagram of restrained area for wind directions: (a) 0, (b) 90, (c) 180 and (d) 270.](image)

**Figure 39:** Magnified shape of the restrained area for wind directions: (a) 0, (b) 90, (c) 180 and (d) 270.

The ellipse effect is better reflected in the shape of the restrained area (Figure 39). However, the restrained area takes a more sophisticated shape than an ellipse, as the wind, terrain and vegetation all vary across the landscape. Figure 39 shows how the variation in vegetation affects the shape of the area. We can observe that except in scenario d, red edges, which are the areas with the least vegetation resistance to fire, are always part of the restrained area. That is because the fire travels faster in that area. However, in scenario d, the wind direction is in favor of the source-destination direction, which makes the non-dominated paths so narrow that the fire will most probably reach the destination before even entering the red area.
In addition to vegetation, factors such as aspect, slope and wind also affect the fire direction and speed. The shape of the restrained area reflects the effect of all these factors on the fire most probable travel paths.

6.6.2 Distribution of the fire traversal time

The fire traversal time distribution is derived by aggregating the output of the simulation. To determine the simulation stability period, multiple $m$ values have been evaluated to assure the capture of chance-related signals. Any number greater than 20 resulted in the desired one percent accuracy level.

In order to evaluate the accuracy of the estimation, the empirical distributions from the reduced-size network are compared with their counterparts from the original network. In Figure 40, the comparison is graphically displayed by overlaying the distributions. In each scenario, the two histograms are in more agreement with each other as their plots become more closely aligned.

The $Q-Q$ plot for the histograms in Figure 40 are presented in Figure 41, which better display the difference in shape, location and skewness of the distributions.
Figure 40: Comparison of the fire arrival time empirical distribution of the reduced-size network versus the distribution derived from the original network.
Figure 41: Q-Q plot of the fire arrival time empirical distribution of the reduced-size network versus the distribution derived from the original network.

In addition, we use the KS nonparametric test to check whether the two empirical distributions in each scenario are drawn from the same distribution. The results are shown in Table 3 in which the KS-statistic values are compared with the critical value of 0.192 at $\alpha = 0.05$. 
Table 8: KS-Statistic values for the four scenarios compared with the critical value.

<table>
<thead>
<tr>
<th>Wind Direction</th>
<th>KS-Statistic</th>
<th>Hypothesis Conclusion</th>
</tr>
</thead>
<tbody>
<tr>
<td>0°</td>
<td>0.074 &lt; 0.192</td>
<td>Fail to Reject</td>
</tr>
<tr>
<td>90°</td>
<td>0.010 &lt; 0.192</td>
<td>Fail to Reject</td>
</tr>
<tr>
<td>180°</td>
<td>0.0053 &lt; 0.192</td>
<td>Fail to Reject</td>
</tr>
<tr>
<td>270°</td>
<td>0.006 &lt; 0.192</td>
<td>Fail to Reject</td>
</tr>
</tbody>
</table>

Looking at the validation results in Figure 40, Figure 41 and Table 8, we can observe that the distributions from the reduced-size network are strongly congruent with those of the original network. In other words, in the four scenarios the methodology is proved to be effective in reducing the simulation run-time, and accurate as compared to the simulation results from the original network.

6.7 Value of the reduction methodology

As noted before, any reduction in the network size directly impacts the simulation run-time. We have selected the number of nodes to represent the network size. The original network consists of 5733 nodes.

Table 9: Effect on wind direction on the number of Step 1 nodes.

<table>
<thead>
<tr>
<th>Wind direction (degrees)</th>
<th>Original network node count</th>
<th>Node reduction output</th>
<th>Path enumeration output</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>%Reduction from original network</td>
<td>%Reduction from original network</td>
</tr>
<tr>
<td></td>
<td>Node count</td>
<td>%Reduction from original network</td>
<td>Node count</td>
</tr>
<tr>
<td></td>
<td>2819</td>
<td>50.8</td>
<td>783</td>
</tr>
<tr>
<td></td>
<td>2815</td>
<td>50.9</td>
<td>552</td>
</tr>
<tr>
<td></td>
<td>3064</td>
<td>46.5</td>
<td>713</td>
</tr>
<tr>
<td></td>
<td>784</td>
<td>86.3</td>
<td>138</td>
</tr>
</tbody>
</table>
Table 9 shows the relative size of the network from each step of the size-reduction methodology for all the scenarios.

![Simulation Run-Time Comparison](image)

**Figure 42:** Monte-Carlo implementation time on the reduced-size network compared with the implementation time on the original network.

Figure 42 illustrates the difference between the simulation run-time of the original and reduced-size networks, resulting from the size reduction. On the average, a 58.75 percent reduction in simulation time is observed which is a notable achievement for the methodology.

### 6.8 Multiple ignition points

One might argue that it is very unlikely to spot the fire at its ignition time and most of the time a fire is detected when it has already spread to the neighboring areas. Also, fire may start in multiple places at the same time. On the other hand,
the area of interest may span over multiple network nodes, and so a single point of interest is not an accurate representation. The algorithm is capable of easily handling all of these cases. In case the ignition area spans over multiple points, we first find the intersection of the corresponding fire perimeter area with network edges. Next, new network nodes are introduced at the intersection points. Lastly, all the intersection points are connected via zero traversal time edges to a dummy node, which will be fed into the algorithm as the only ignition point. Right after Dijkstra’s algorithm processes the dummy node, all the connected intersection nodes will be visited simultaneously.

![Figure 43: Multiple ignition points, multiple destination points and already propagated fires are handled using dummy node replacement](image)

Multiple ignition areas/points are treated in the same way. We also use the same approach to handle multiple points of interest, except in that case, the dummy node will be introduced as the only point of interest to the algorithm. Figure 43 illustrates a case in which the above three scenarios are presented.
6.9 Stochastic versus deterministic analysis

Stochastic analysis of wildfire has a significant advantage over the deterministic approach. In the deterministic approach, only one single number is presented as the expected time of fire travel time. This number does not convey any information about the reliability of this prediction and the actual fire arrival time will almost never be the same. However, with stochastic analysis, instead of a single number, we provide a probability distribution of the arrival time. This distribution enables us to provide answers to many previously unanswered questions such as:

- What is the probability that the fire reaches a specific area within an hour?
- How much time do we have until the fire hits an area with 90 percent probability?
- What is the expected arrival time of the fire to an area and what is the likelihood of fire arrival before that time?

Another advantage of the stochastic analysis is that the expected minimum source-destination fire traversal time in a network with edges of uncertain time lengths is always shorter than the deterministically calculated source-destination traversal time using the average time lengths of edges. In other words, the fire is expected to hit the area of interest, earlier than the fire arrival time derived based on a deterministic network. The proof is as follows.
Proof:

Let \( G(V, E) \) be the graph network where the edges are associated with positive continuous random variables \( E_{ij} \sim F(.) \) CDF with \( E[E_{ij}] = t_{ij} \), where \( E_{ij} \) are independent. Let \( P^0 \) be the deterministic shortest path from \( s \) to \( t \) with its associated travel time random variable as \( T^0 \). Thus its length is \( T^0_E = \sum E[E_{ij}] = \sum t_{ij} \) where the summation is over the edges belonging to the shortest path \( P^0 \).

Lemma 1. Let \( Y_i \) be positive random variables \( \sim F_i(.) \), \( i = 1, 2 \).

then:

\[
E[min(Y_i)] \leq \min(E[Y_i])
\]

Proof: Let \( W = \min(Y_1, Y_2) \). Then

\[
P(W > w) = P(Y_1 > w) \cdot P(Y_2 > w)
\]

\[
= (1 - F_1(w)) \cdot (1 - F_2(w)).
\]

For positive continuous random variables we have \( E[X] = \int (1 - F(x))dx \). Thus:

\[
E[W] = \int (1 - F_1(w)) \cdot (1 - F_2(w)) \leq
\]

\[
\int (1 - F_1(w)) = E[Y_1] since(1 - F_2(w)) \leq 1 for anyw.
\]

By the same argument it follows that \( E[W] \leq E[Y_2] \). Thus: \( E[min(W)] \leq \min(E[Y_i]) \) for \( i = 1, 2 \).
Theorem 3. $E[T^0] \leq T^0_E$, i.e. the expected time of the stochastic shortest path is less than or equal to the deterministic shortest path (based on the expected time of the edges).

Proof.

(i) If only one path exists from $s$ to $t$, then clearly $E[T^0] = T^0_E$.

(ii) Suppose there exist two paths, $P^1$ and $P^2$ with $T^1$ and $T^2$ random variables. If disjoint, then $T^1 = \sum [E_{ij}]$ where summation is over the edges in $P^1$ and $T^2 = \sum [E_{ij}]$ where summation is over the edges in $P^2$.

Here, $T^1$ and $T^2$ are independent random variables with cdfs $F_1(.)$ and $F_2(.)$ and from Lemma 1: $E[min(T^1, T^2)] \leq min(E[T^1], E[T^2]) = \text{ the length of the deterministic shortest path}$.

(iii) Assume there exist only two paths $P^1$ and $P^2$ but they are not disjoint (i.e. they share some edges). Let $E^s$ be the set of all common edges $P^1$ and $P^2$, i.e. $P^1 \cap P^2$. Also let $P^{1s}$ be a path consisting of all the edges that exist in $P^1$ but not in $E^s$ i.e. $P^1 \setminus E^s$. Similarly $P^{2s}$ for $P^2 \setminus E^s$. We have:

$$E[min(T^1, T^2)]$$

$$= E[min(T^{1s} + T^s, T^{2s} + T^s)]$$

$$= E[min(T^{1s}, T^{2s}) + T^s]$$

$$= E[min(T^{1s}, T^{2s})] + E[T^s] \leq min(E[T^1], E[T^2]) + E[T^s]$$

$$= min(E[T^1] + E[T^s], E[T^2] + E[T^s])$$

$$= min(E[T^1], E[T^2])$$
the length of the deterministic shortest path, where again we used Lemma 1.

(iv) The proofs in (ii) and (iii) can be easily extended by induction to the case with more than two paths.

6.10 Conclusion

Wildfires destroy about 4 to 5 million acres of land in the US per year, contributing to worsening air pollution and leading to innumerable preventable injuries and fatalities. Fire prediction models can help in firefighting efforts to control the wildfire spread and therefore in saving lives and property. The degree of accuracy in fire prediction models depends on the model’s applicability to a given situation and the reliability of the model output.

In fire propagation, wind is one of the most influential yet unpredictable determinants. In this Chapter we presented a model that accounts for the variability of the wind by modeling its effect as a random variable, making it applicable to most of the real-life wildfire events. Also, the model is capable of providing reliability measures for the output in terms of probability values. Wildfire propagation was modeled using the Stochastic Shortest Path problem with the objective of deriving the distribution of fire arrival time to an area of interest given the location and time of the ignition. We chose a Monte-Carlo implementation of the Dijkstra’s algorithm to find the fire travel time distribution as there are no exact methods for its derivation in a timely manner.
In the event of a fire, the model can generate outputs to allow enough time to plan fire suppression and population evacuation activities and to react to changes in environmental conditions real-time. In order to decrease the simulation time, we proposed a network size reduction methodology with the ability to identify and eliminate parts of the network that have negligible effect on the fire travel time distribution. The implementation of the reduction methodology was demonstrated to be effective in reducing the simulation runtime by 58.75 percent on average and at the same time providing a highly accurate output compared with the simulation results on the original network.

This methodology can be utilized in any application involving the calculation of stochastic shortest time, such as road networks and telecommunication networks although it is most suitable for the case of wildfire due to the characteristics of fire propagation. The fast implementation time of the algorithm to facilitate what-if analysis and perform recalculations in near real time makes it especially valuable.

For future work it is possible to relax the assumption of constant wind direction considered in this study. One way to include the variability of the wind direction is to consider it as a time-dependent factor that changes according to the forecast during the course of simulation. Another possible way is to introduce a second random variable into the fire ROS equation to account for the wind direction variations.

Another potential improvement to our model is to devise a smarter way of network construction to provide more realistic fire paths in each direction. One
way of doing this is by introducing extra random paths in the direction of the point of interest.
7. Summary and future research

In this research we studied the wildfire propagation behavior and provided a stochastic framework for predicting the wildfire spread on a heterogeneous landscape. A methodology based on Delaunay triangulation was utilized to tessellate the heterogeneous landscape to multiple homogeneous sub-regions and then construct a network representing the fire propagation paths. Wind speed was identified as one of the most important factors affecting the fire rate of spread. In order to capture the variability of the wind speed, a random variable with the Weibull distribution was used to represent the wind speed. Rothermel’s equations were then utilized to derive the fire rate of spread distribution based on the wind speed distribution.

We introduced the concept of stochastic wildfire contours to predict the fire front-line considering the fire rate of spread distribution. These contours enable the fire predictor to evaluate the reliability of the prediction and make the appropriate decision accordingly. The methodology utilized for deriving the fire stochastic contours was adopted to generate another set of contours, introduced as the stochastic evacuation contours. These contours give information on how close the fire is and how probable it is for the fire to reach the point of interest in a specific time.

In another Chapter we studied the problem of finding the fire traversal time distribution from its ignition point to a point of interest. A Monte-Carlo of the depth-limited shortest path was employed for estimating the distribution. We proposed a novel two-step methodology to reduce the run-time of the simulation
by eliminating the parts of the network that have negligible contribution to the estimation error of the Monte-Carlo output. The results from the case study proves the reduction methodology to be effective, being able to reduce the runtime of the simulation by 58% on average. This reduction is achieved without affecting much the accuracy of the wildfire traversal time distribution, which is examined by the Kolmogorov-Smirnov test.

The work in this research can be extended in different ways. First, variability can be introduced in more factors influencing the wildfire spread. Although during the course of a few hours most of the other factors are expected to remain constant, huge wildfires may keep burning the area for days or even weeks. This will result in change of factors such as humidity, temperature or even vegetation. The most obvious option for further consideration is wind direction, which is also prone to sudden changes in some cases.

Furthermore, time dependency can be introduced into the model, which is especially effective for predictions of further future of propagation. For example, the parameters of the wind speed distribution will most likely change if the propagation spans over days or weeks. Although it is still possible to take advantage of the current model by re-evaluating the parameters each time there is a significant change, the current model is not able to account for the change of parameters in advance. Finally, this model can be incorporated into an evacuation routing system to improve the routing decision. Most of the current evacuation routing models consider the fastest evacuation paths without much regard to the future path of fire. This may result in the fastest route not necessarily being the safest route or even a feasible route. An integrated model
will be able to suggest the fastest evacuation routes accounting for the possible future obstruction of the roads because of the fire.
REFERENCES


Linn, R. R. (1997). *A transport model for prediction of wildfire behaviour*. (PhD), New Mexico State University, Los Alamos National Laboratory.


