SELECTING LOCATIONS FOR FOREST FUEL TREATMENTS USING SIMULATION OPTIMIZATION

by

Adam Rytwinski

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science in Forestry

Faculty of Forestry and the Forest Environment
Lakehead University
2009

ABSTRACT

Rytwinski, Adam. (2009). Selecting locations for forest fuel treatments using simulation optimization. 102 pp.

Keywords: fuel management, operations research, simulation optimization, wildfire, stochastic fire simulation.

Wildfire in the Canadian boreal forest poses a considerable threat to forest inventory, structures and human lives. Traditional reactive methods of fire suppression fail to mitigate this threat and can worsen the problem by causing unnatural fuel buildup. Fuel management, carried out through the application of fuel treatments across a landscape, is a proactive approach to fire hazard reduction which has the potential to decrease fire risk significantly. The problem of selecting the location of fuel treatments is complicated by the uncertainty of predicting when and where fire events will occur.

In this study, the problem of how to locate fuel treatments across a forested landscape is formulated as a combinatorial simulation optimization problem with the objective of minimizing fire risk while remaining within a budget constraint. A spatial computer simulation of fire spread, FastFire, is designed and validated to quantify fire risk. The simulation optimization formulation is tested on a 220,000 ha forested area in northern Ontario. A set of 100 potential fuel treatments are designed and the goal of the optimization is to choose a subset of treatments. The optimization problem is solved using the OptQuest optimization engine and the resulting solutions are shown to perform 5% better than solutions chosen using a greedy heuristic. The spatial layout of the best solutions are visually analyzed using GIS software to understand trends that resulted in lessened fire risk.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

Thanks to: Kevin Crowe, Dave Savage, Rob Luik, Tomislav Sapic, David Martell and the Sustainable Forest Management Network, Barry White and the Alberta Sustainable Resource Development, family and friends.

A CAUTION TO THE READER

This M.Sc.F thesis has been through a semi-formal process of review and comment by at least two faculty members.

It is made available for loan by the faculty for the purpose of advancing the practice of professional and scientific forestry.

The reader should realize that opinions expressed in this document are the opinions and conclusions of the student and do not necessarily reflect the opinions of either the supervisor, the faculty or the University.

1. INTRODUCTION

1.1. Significance and Nature of the Fuel Management Problem

Wildfire in the Canadian boreal forest poses a significant threat to lives, infrastructure, and forest resources when large fires escape suppression and burn out of control (Hirsch, Kafka and Todd 2004; Omi 2005; Vicars 1999; Armstrong and Cummings 2003; Palma *et al.* 2007). Between 1986 and 2005 there have been 36 deaths directly related to wildfire in Canada (Natural Resources Canada 2008) and 133 deaths in the U.S. between 1990 and 1998 (Mangan 1999). Despite national spending on fire suppression on the order of $400 to $800 million per year (Canadian Forest Service 2007) the significant damage caused each year by forest fires in Canada persists. For example, in B.C., in the summer of 2003 an estimated $700 million dollars worth of damage was done, including the destruction of 334 homes and the evacuation of over 45,000 people (Filmon 2004). In 2001, one fire alone in Chisholm, Alberta caused the forest industry to lose over 4.5 million cubic meters of growing stock and 6,300 hectares of regenerated stands (Dessorcey 2001). Because there is such a potential for massive devastation, the large majority of fires in Canada are suppressed before they escape to a unmanageable size.

One unintended consequence of aggressive fire suppression is that it prevents low intensity fires that would reduce fuel buildup, and thus facilitates an increase in fuel loads which, in turn, results in a higher probability of disastrous, uncontrollable fires occurring (Conrad, Hartzell and Hilbruner 2001; Agee and Skinner 2005). Several researchers have observed that (van Wagner 1988; Skinner *et al.* 1999; Skinner *et al.*

2002; Podur, Martell and Knight 2002; Stocks *et al.* 1998) in recent years there has been an upward trend in the annual amount of forest burned by wildland fires in Canada. Flannigan *et al.* (2005) speculated that, as a result of climate change, the area burned in Canada could double by the end of the century (Flannigan *et al.* 2005). For all of these reasons, a more proactive approach to fire management, with greater emphasis on controlling the build-up of forest fuels, is growing in importance.

Fuel management is typically defined as the process of altering the amount and structure of fuels in a forested landscape in order to reduce the spread and intensity of wildfires *before* they occur (Pyne 1984; Finney 2001). It requires a strategy developed at the landscape-scale, which is then applied at the stand level. Such treatments vary according to management objectives, fuel types, proximity to urban areas and other factors.

The treatment of forest fuels can be divided into three categories: reduction, conversion and isolation (Pyne 1984). Reduction is concerned with removing high-risk fuels within a forest stand. Usually surface-fuels, such as shrubs and downed woody debris, are removed, along with ladder fuels that provide a vertical pathway to the forest's canopy. The objective of reducing surface fuels is to minimize the intensity and spread of surface fires; the objective of removing ladder fuels is to reduce the opportunity for surface fires evolving into crown fires which have a much greater potential for destruction. Fuel reduction is typically achieved through prescribed burning or harvesting operations (Agee and Skinner 2005; Martell *et al.* 2004).

The second major approach of treating fuels (conversion treatments) replaces stands of highly flammable fuels with fuels that are more resistant to fire and burn with

less intensity (Pyne 1984). One conversion approach used in Canada involves converting conifer and mixed-wood stands to pure aspen stands to act as a natural defense against wildfire spreading to First Nations communities (Dallyn *et al.* 2000).

Fuel-isolation is a third approach to managing fuels; this involves disrupting the continuity of highly flammable fuels and separating valuable resources from high risk fuels through the creation of fuel-breaks (Agee *et al.* 2000). A fuel break is defined as a strategically placed strip of land in which vegetation has been removed to restrict the spread of fire (Green 1977). Agee *et al.* (2000) conclude that well designed fuel-breaks in conjunction with other fuel treatments are essential for intelligent fuel management.

While there exists a variety of methods for treating fuels, each having distinct effects on wildfire behaviour, it is important to understand that for a method to be effective, the layout of all treatments as a whole must be considered. Since large wildfires cover an area greater than that of a typical treated stand, a single fire could potentially encounter several fuel treatments before reaching extinguishment. Hence it has been conjectured, based on modeling and speculation, that a set of fuel treatments within a forest in which the location of each treatment has been considered in relation to all other treatments could provide significantly greater protection than randomly or greedily chosen solutions (Finney 2001; Hof and Omi 2003; Martell *et al.* 2004). In other words, the whole is greater than the sum of its parts when the spatial location of all treated stands are considered, not in isolation from one another but as a whole. Henceforth we will refer to the spatial layout of all treated areas throughout a forest as a *fuel treatment mosaic.*

At present, large scale fuel management plans are seldom used in the boreal forests of Canada for several reasons. First, there is inadequate empirical evidence to support the conclusion that fuel treatments are cost effective (Carey and Schumann 2003). This is not surprising; for the cost and danger associated with deliberate burning of forests to study fire behaviour is a major constraint upon such empirical enquiries. There has only been one large-scale controlled experiment examining how fire behaves when coming in contact with a treated area, the International Crown Fire Modeling Experiment involving 18 high-intensity crown fires in the Northwest Territories (Alexander and Lanoville 2004). While the results were promising, they were far from conclusive (Schroeder 2006).

Many studies have either focused on examining the aftermath of naturally occurring fires that burned into treated areas (e.g. Pollet and Omi 2002; Finney 2005) or have relied on computer simulation (e.g. van Wagtendonk 1996; Stephens 1998; Fule *et al.* 2001). While the literature appears to justify the use of fuel treatments, many researchers still call for more experimental and site-specific evidence (*e.g.* Graham, McCaffrey and Jain 2004; Carey and Schumann 2003; Fernandes and Botelho 2003). This explains the second reason that fuel treatments are seldom used; namely, that the *cost* of treating large areas of forested land can be impractical (Finney 2003; Lynch 2003) and managers are therefore reluctant to invest a large amount of money into anything with uncertain results. This uncertainty is increased because of the lack of any decision support software that could aid land managers in choosing an efficient fuel management strategy or provide an accurate figure for the amount of money that could be saved. The goal of this study is to address this lack by proposing and testing a

decision support tool capable of accurately designing and evaluating effective fuel treatment mosaics.

1.2. Challenges in Modeling this Problem

The design of effective fuel treatment mosaics presents an interesting challenge for decision support modeling. According to Martell (2007):

> Fuel management represents rich untapped sources of interesting problems for operations research (OR) specialists, particularly those interested in spatially explicit stochastic integer programming under uncertainty... There is a very rich source of interesting problems that will push even the best stochastic integer programming methodologies to their limits.

In operations research, the fuel management problem can be formulated as a minimization problem.

[1] $$minimize\ Z = D(\boldsymbol{x})$$

subject to:

[2] $$\boldsymbol{c}^T\boldsymbol{x} < B$$

Where $\boldsymbol{x}$ is a vector of binary decision variables representing a proposed fuel treatment mosaic, $\boldsymbol{c}^T$ is a transposed vector representing the cost of each treatment in the mosaic, B is a real number representing a budget constraint, and D is a function that quantifies

fire risk. In other words, the solution to this model represents selected treatments that are within the budget constraint while minimizing the risk of fire damage throughout the landscape.

Researchers who have attempted to solve this problem have consistently run into the same problem: wildfire behaviour is extremely complex and stochastic. Hence, calculating fire risk in a forest is not a trivial task (He *et al.* 2004; Finney 2005). In general, the accurate quantification of fire risk requires a spatial fire computer simulation program (Finney 2005). These programs, e.g. Prometheus (Tymstra 2002), LANDIS (He, Mladenoff and Crow 1999) and FARSITE (Finney 1998), are often also highly complex, require large amounts of input data, and can take significant computational time to assess the risk of one potential mosaic of fuel treatments. The problem with this complexity, from an optimization standpoint, is that the calculation of fire risk is required by the model's objective function (see [1] above). Even moderate sized optimization problems typically require in the order of thousands or more objective function calculations (Hilier and Lieberman 2005), so an increase of several seconds per objective function calculation will increase the total optimization time substantially. Hence, a simulation model used must be optimized for speed if there is to be any chance of making this problem feasible.

Because of the high cost of this fundamental computational problem, two approaches to modeling this problem have prevailed:

i. design models that simplify the process of fire risk calculation (e.g. Hof and Omi 2003; Wei *et al.* 2008) in order to facilitate optimization, or

ii. design stochastic simulation models to support decisions on the problem (e.g. Finney 2001; Parisien, Junor and Kafka 2006).

The trouble with the former approach is that the complexities of fire behaviour are not easily captured in traditional optimization methods. Therefore, doubts over the realistic utility of these solutions are justifiable (Hof and Omi 2003). The problem with the latter approach is that only a handful of the great amount of potential fuel treatment mosaics are examined and no computerized guided search can be employed to find high quality or near optimal solutions.

An alternate approach to modeling this problem is to use *simulation optimization.* Simulation optimization is the process of finding the best configuration of decision variables for a given system where the performance is evaluated based on the output of a computer simulation model of the system (Gosavi 2003). The optimization itself is an iterative process of choosing a solution which is then passed to a simulation model for evaluation and subsequently passed back to the optimization model which will choose the next solution based on the results. The use of simulation optimization as a method of solving complex optimization problems has seen a remarkable growth in recent years both in the academic world and in practice (April *et al.* 2003). Optimization via simulation has been successfully applied in wide range of areas including logistics (e.g. Hill and Fu, 1994), supply chain management (e.g. Azadivar *et al.* 1996) and manufacturing (e.g. Morito *et al.* 1993; Vogt 2004).

The idea of employing simulation optimization for spatially allocating fuel treatments has already been proposed by Hof and Omi (2003) and Martell (2007); yet both

agree that the computational time required to evaluate each potential solution poses a significant computational challenge.

1.3. Objective and Outline of this Research

The central objective of this study is to evaluate whether simulation optimization can be used to create a decision support tool capable of designing efficient fuel treatment mosaics in real forested landscapes. The research is outlined as follows. A literature review focuses on two main areas. Firstly, the literature on the prediction of fire behaviour is reviewed. The second section includes literature concerned with methods of solving simulation optimization problems. In methods, a formulation of the fuel management problem in a simulation optimization context is present, as well as the design and calibration of a spatial fire simulation model, and the integration of OptQuest® software as a method of solving the simulation optimization problem. The case study presents a fuel treatment allocation problem set based on a forest in the boreal forest of Northwestern Ontario. The results section summarizes the calibration of the simulation model and will present the solutions found for the case study problem set using simulation optimization and compare these with other methods. The Discussion addresses the strengths and weaknesses of the simulation optimization approach as well as the implications the findings may have on modeling the harvest-scheduling problem.

2. LITERATURE REVIEW

The objective of this research requires a review of the literature in two main areas. First, methods of modeling fire behaviour and quantifying fire risk are outlined. This background was necessary for the design of a computer simulation model of fire spread used in the simulation optimization framework. The second section reviews research in the field of simulation optimization with a focus on formulating and solving combinatorial optimization problems.

2.1. Modeling the Behaviour of Fire

Research into the design of accurate predictive models of wildfire behaviour in North America dates back to the 1920s (Stocks *et al.* 1998). Such models have since been developed and refined for diverse purposes, including planning for fire management activities (van Wagtendonk 1996), modeling forest landscape change (Gardner, Romme and Turner 1999) and evaluating fuel treatment prescriptions (Finney 2001).

Fire behaviour models represent a class of mathematical models concerned with the prediction of how a wildfire will behave given a fuel type and a set of weather conditions; but such models do not attempt to predict the spatial extent or progress of the fire. Models that predict the spread of fire are an extension of fire behaviour models, using their parameters to predict how a fire will progress spatially and temporally through a heterogeneous landscape.

Models of fire behaviour are designed to predict and describe parameters such as rate of spread, fire intensity and fuel consumption using inputs based of fuels, weather and

topography (Perry 1998). Models of fire behaviour typically fall under one of three categories: physical, semi-physical and empirical.

Physical models are those based purely on mathematical descriptions of the physical and chemical processes of fire events. According to Catchpole and de Mestre (1986), a physical model of fire predicts fire-behaviour directly from the analysis of the chemistry of combustion. Such a model would theoretically be the most desirable approach to the prediction of fire behaviour since it is based on known relationships and hence could easily be scaled to different fuel types and weather conditions (Perry 1998). While several theoretical physical models exist (e.g. Thomas 1967; Weber 1991), the difficulty of quantifying the combustion and heat transfer through fuel environments as varying and complex as a typical forest stand, has made the creation of an operational physical model impossible (Perry 1998).

Semi-physical approaches, combining both physical and empirical techniques, have also been developed. These models still incorporate mathematical descriptions of heat-flux and ignition, but derive the various constants governing these equations experimentally, through observations of test-fires (Perry 1998). These observations are then extrapolated to predict the behaviour of fire at larger scales. Currently, the most widely used fire behaviour model is a semi-physical model developed by Rothermel (1972) which forms the basis of the National Fire Danger Rating System (Andrews 1986), the BEHAVE fire prediction system (Burgan and Rothermel 1984), and the FARSITE fire area simulator (Finney 1998) in the USA. The Rothermel equation for the rate of spread through surface fuels is based on the conservation of energy and predicts rate of spread (ROS) as:

[3] $$ROS = \frac{Heat\ flux\ absorbed\ by\ a\ unit\ volume\ of\ fuel}{Heat\ required\ for\ ignition\ of\ that\ volume\ of\ fuel}$$

The rate of spread equation has been used in many different environments from North American grassland fuels (Sneeuwjagt and Frandsen 1977) to garrique in southern France (Malanson and Trabaud 1988). Several problems have been reported questioning how well the model predicts the behaviour of fire in structurally complex fuel systems (van Wilgen and Richardson 1985). Gould, for example (1988), experienced problems when applying the model in the presence of high wind-speeds.

Empirical models do not include any of the physical elements of fire spread, but instead employ statistical descriptions of observed wildfires to predict how future fires would behave. Through observing the effect of elements such as fuel type, wind speed and fuel moisture on the behaviour of observed wildfires, equations can be formed to predict the behaviour of fires under many fuel and weather conditions. These models are relatively easy to construct and have been used to support the McArthur fire danger meter in Australia (Noble, Barry, and Gill 1980) and the Canadian Forest Fire Danger Rating System (Stocks *et al.* 1998). Although these models have been highly successful at predicting fire behaviour within the conditions and ecosystem for which they were designed (Weber 1991), extrapolating results into other regions has been unsuccessful (Marsden-Smedley 1993). It is therefore recommended that the use of empirical models outside of test conditions be treated with caution (Chandler *et al.* 1983).

An overview will now be given of two of the most prominent fire behaviour prediction models in North America: the Fire Behavior Prediction and Fuel Modeling System (BEHAVE) and the Canadian Forest Fire Danger Rating System (CFFDRS).

2.1.1. The Fire Behavior Prediction and Fuel Modeling System (BEHAVE)

The BEHAVE system is a semi-physical approach to modeling the behaviour of surface fires for fuel types found in North America. It is comprised of two subsystems: FUEL and BURN. The FUEL subsystem classifies fuel types and provides thirteen standard fuel models which may be used unaltered or modified to create new fuel models (Burgan and Rothermel, 1984). The BURN subsystem predicts rate of spread and frontal fire intensity for ground fires based on Rothermel's (1972) model (Andrews 1986). The model requires weather and topography data as well as a fuel model as described by the FUEL subsystem. BehavePlus is a modified version of the BEHAVE system designed to run on modern computers and to incorporate other aspects of fire behaviour, such as crown fire, spotting and tree mortality (Andrews *et al.* 2005).

2.1.2. The Canadian Forest Fire Danger Rating System (CFFDRS)

The CFFDRS is a national system of rating the danger of forest fires and uses an empirical approach to predicting fire behaviour. Although the CFFDRS is classified here a a behaviour model, it does include a secondary output the roughly predicts fire perimeter size which would be considered a model of fire spread. The current form of the CFFDRS has been in development since 1968 although it incorporates research dating back as far as 1925 (van Wagner 1987; Stocks *et al.* 1998). It is composed of two subsystems: the Fire Weather Index System (FWI) and the Canadian Forest Fire Behaviour Prediction System (FBP).

The purpose of the FWI is to predict the danger of forest-fire across Canada by estimating forest fuel moisture content. Inputs to the FWI include temperature, relative

humidity, wind and rain. Outputs are Fine Fuel Moisture Code (FFMC), Duff Moisture Code (DMC) and Drought Code (DC). These codes are then used to calculate Initial Spread Index (ISI) and Buildup Index (BUI) which offer numerical measures of the potential for fire intensity and fire spread . Finally these indices are used as inputs for the FBP. Calculations for the FWI indices rely solely on daily weather readings of relative humidity, wind speed, 24-hour accumulated precipitation and dry bulb temperature taken at the typical peak burning time of 4 p.m. (Stocks *et al.* 1998). For a detailed description of the FWI see van Wagner (1987).

The second component of the Canadian Forest Fire Danger Rating System, the Fire Behaviour Prediction System is a method of predicting fire behaviour based on fuel type, weather and topography. The four primary outputs of the model are:

i. Rate of Spread (ROS),

ii. Head Fire Intensity (HFI),

iii. Fuel Consumption and

iv. fire description code.

The system also includes eleven secondary outputs which describe back and flank spread rates and give an approximate fire area and perimeter based on a simplistic elliptical model of fire spread in homogeneous fuel types (see Alexander 1985). The effect of slope on the rate of spread is also taken into account (van Wagner 1988). Canadian fuel types were divided into sixteen categories representing the most common fuels found in Canada - ranging from grasslands to mixed wood forests to slash piles. The FBP is an empirical

model based on data collected from 409 experimental fires and 86 documented wildfires. For a detailed description of the FBP see Hirsch (1996).

The CFFDRS is a reliable and effective model of fire behaviour designed specifically for use in Canadian forests. Reports on the accuracy of the system have shown excellent agreement between observed versus predicted values (e.g. Stocks and Flannigan 1987; Stocks 1988; Hirsch 1989). The FBP system forms the basis of many prominent fire spread models including Prometheus (Tymstra 2002), BFOLDS fire model (Perera *et al.* 2002) and Sem-land (Li, Ter-Mikaelian, and Perera 1997). In the following section, the prediction of spatial fire spread is described.

2.2. Models of Fire Spread

Models of fire spread predict the spatial direction (and therefore pattern) of fires burning through a forest. They are used to predict which areas of the forest are most likely to be affected by fire. The purposes these models serve can include guiding the suppression of fires (e.g. Ntaimo *et al.* 2004) or predicting forest landscape successional change over time (e.g. He and Mladenoff 1999). Models of fire spread can also be classified as either *stochastic* or *deterministic* (Finney 1999). This classification is independent of the conceptual model employed. By definition a deterministic model will return identical results each time it is run with the same initial conditions. Stochastic models incorporate probabilistic elements and identical initial results do not imply identical results. Choosing whether to take a deterministic or stochastic approach depends on the modeling objectives. In general, deterministic models are better suited for highly accurate and small scale predictions of fire behaviour which could, for example, guide the

deployment of fire suppression personnel (Hargrove *et al.* 2000; He and Mladenoff 1999). Stochastic models, on the other hand, are typically superior for applications at a larger scale such as predicting fire risk across a landscape (He and Mladenoff 1999).

In this section the three most common conceptual models employed in modeling fire spread are described:

i. shape models;

ii. vector models; and

iii. raster (or cellular) models.

2.2.1. Shape Models

Shape models are the simplest models used to predict the spread of a forest fire. They were originally developed for estimating the growth of a fire under uniform environmental conditions (e.g. Fons 1946). The simplifying assumption of these models is that fires burn in spatially uniform conditions with unchanging weather and will assume a specific shape. Under this assumption, given an ignition point and an estimate of the forward spread rate, changes in the fire's size could be estimated simply as a function of time (Finney 1998). The elongated ellipse is often assumed to be the best fit shape for fires burning in heterogeneous conditions (Green, Gill, and Noble 1983) although this assumption has been challenged by the observation that specific fires have been shown to be better described as egg-shaped, ovoid or double ellipse (Peet 1967; Albini 1976; Anderson 1983).

The appeal of shape models is their simplicity. In general, only three inputs are required: fire front intensity, elliptical shape factor (length-to-breadth ratio) and the

backfire rate of spread (Gardner, Romme, and Turner 1999). The major downside to this type of model is that real-world fires rarely burn in uniform conditions and that the predictive powers of the model are markedly decreased when applied to heterogeneous beds of fuel (Perry 1998). True fires constantly encounter changes in wind, elevation and fuel type and therefore require a more realistic model capable of dealing with such variations.

Despite the drawbacks of this conceptual model, elliptical fire models still play an important role in predicting the spread of fires. The methods described by Alexander (1985) assume an elliptical fire shape in order to predict the flank rate of spread when given a frontal and backing spread rate. This method is employed as part of CFFDRS (Hirsch 1996) and is used in certain vector models to estimate small fire shapes.

2.2.2. Vector Models

Vector models represent the most recent advances in simulating the spread of fires and are arguably the best approach for dealing with spatial and temporal heterogeneity (Finney 1998). Vector models are characterized by their use of infinitely thin arcs connecting a set of vertices in a Cartesian plane to represent the fire's front as it expands over time (Finney 1998; Keane *et al.* 2004). Although it is not a defining property, currently all existing vector models rely on Huygens' principle of wave propagation (Morais 2001). Hence, vector models are often referred to as "wave" models. The fundamental assumption of wave models is that Huygens' principle, which was originally intended to describe the propagation of light waves, is analogous to the spread of a fire's front. Huygens' principle states that an advancing wave may be estimated from the sum of

secondary waves arising from individual points in the medium being traversed. When this principle is applied to fire-spread, an expanding fire front can be thought of as the combination of individual fire fronts emanating from points along the fire's perimeter. Figure 1 shows how at each time step in the simulation model, a new fire front is calculated by burning small secondary fires and creating a new perimeter based on their combined shapes.

The method used to calculate the fire fronts at each vertex employs some form of shape model, such as the elliptical model of van Wagner (1969) described above. A cardinal assumption upon which vector models are based is that each defined vertex along the fire front will produce a unique fire pattern based on the environmental conditions at that point, and that the total perimeter of the fire is an amalgamation of these patterns. Vector models allows for a more accurate representation of how fire burns in realistic landscapes (heterogeneous conditions) because of its use of multiple secondary fires, each of which will burn in a pattern depending on the fuel conditions encountered.

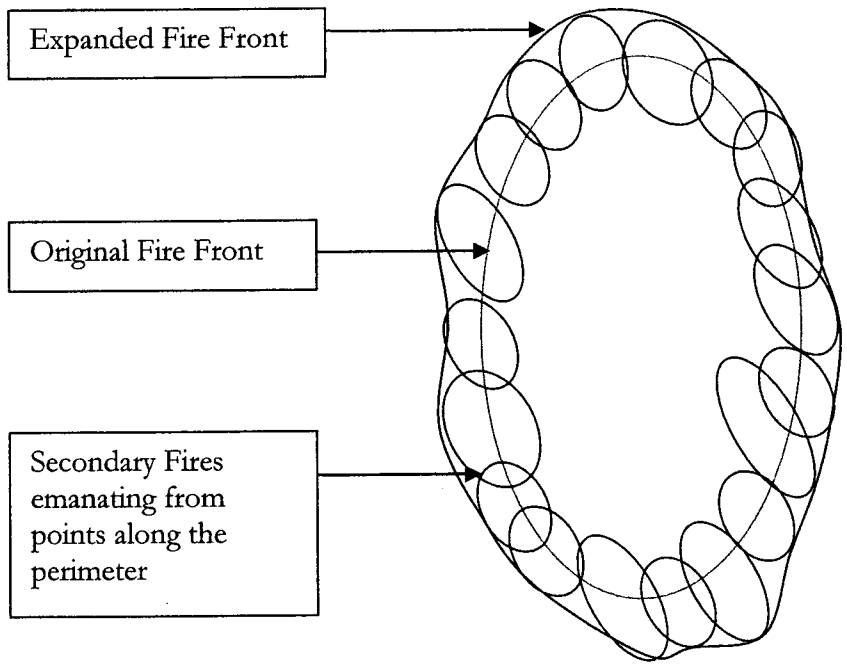

Figure. 1. Fire front modeled as an expanding set of vertices using Huygens' Principle.

While vector models of fire propagation have been shown to perform well at predicting the final shape of wildfires (French *et al.* 1990), their major drawback is that the computational time required to perform even a single fire simulation may be significantly longer than other approaches (Richards 1990; Hargrove *et al.* 2000). For this reason, vector models are used most often when highly accurate predictions of a single fire event is needed, rather than applications that require a large number of fire simulations such as the prediction of fire risk across a large landscape. Vector models based on Huygens' principle form two of the most well known fire models employed in North America: FARSITE and Prometheus.

FARSITE is a deterministic spatial fire growth model, developed by Finney (1998). and based on the fire behaviour model of the American BEHAVE fire system which calculates rate of spread using the Rothermel fire equations for predicting surface fire, while crown fire behaviour is estimated using the models developed by van Wagner (1977 and 1993). The model incorporates many different types of fire spread including surface fires, crown fires and fire spotting (lofted embers igniting fires ahead of the advancing fire front) (Perry 1998). Inputs to the model include raster formatted (data is stored in a row-and-column format and each cell corresponds to a rectangular area of land) fuel and topographical information as well as a stream of weather data (Finney 1998). FARSITE was specifically designed for fuel structures found in the United States and the model has proven less successful when tested in conditions outside of its intended environment (Arca *et al.* 2006).

Prometheus is the Canadian Wildland Fire Growth Model (CWFGM) designed specifically for use within Canadian fuel types (Tymstra 2002). The conceptual model of Prometheus is very similar to that of FARSITE, employing a deterministic wave based approach. The major difference between the two models is that the behaviour model driving the fire spread in Prometheus is based on the Fire Weather Index (FWI) and Fire Behaviour Prediction (FBP) subsystems of the Canadian Forest Fire Danger Rating System. The objective in the design of Prometheus was to predict the hourly or daily growth of wildland fires that have escaped initial attacks. Prometheus is therefore used to support operational and strategic decisions on how to attack a fire front in progress (CWFGM Project Steering Committee 2006). Prometheus is also used as the fire spread module of the BURN-P3 software package which was designed to estimate burn

probability maps for large fire prone landscapes through the simulation of a large number of random fire ignitions (Parisien *et al.* 2005).

2.2.3. Cellular Models

Cellular models of fire spread are characterized by their use of a regularly spaced landscape grid, simulating fire growth as a process of ignitions to neighbouring cells (Finney 1998). Each cell represents an area of the landscape and holds environmental attribute data such as fuel type and topography. The number of possible cells that can be affected by a single burning cell is referred to as the neighbourhood structure of the model. At each iteration of the simulation, each burning cell will spread to a subset of its neighbouring cells in either a deterministic or probabilistic fashion based on the fuel types and weather conditions at each cell. The majority of raster models are based on cellular automata as described by Wolfram (1984), where each cell is in one of a finite number of states and cells are affected only by their neighbouring cells at discrete time intervals. Rules of spread can be deterministic, although many models use some stochastic elements to represent the uncertainty produced by random changes of immeasurable variables such as sudden wind changes.

Cellular models have been known to produce distorted fire shapes because of the limited number of possible directions of fire spread inherent in the use of a fixed neighbourhood spread method; but it is possible to reduce this distortion by increasing the resolution of the gridded landscape, and by using a larger spread-neighbourhood (French, Anderson, and Catchpole 1990). An increased resolution, however, will also have the negative effect of increasing computational time, and much research has gone into finding

the optimal balance between these two factors, i.e. what resolution will minimize distortion without rendering the model useless because of the time required to run a simulation (e.g. Ball and Guertin 1992; Green 1983). Cellular models are generally inferior to wave models at predicting the exact shape of a single fire event (Finney 1998). Cellular models, however, are often used when the goal is to create spatially explicit burn probability (BP) risk maps for large landscapes since they are often less computationally intensive when compared to wave models. BP maps are spatial estimates of the probability of each point on the landscape being affected by fire in the next fire season. These probabilities are estimated from multiple runs of a stochastic fire spread model. BP maps are commonly used for evaluating fire risk to specific locations in a heterogeneous landscape (Gardner, Romme and Turner 1999).

Cellular models of fire spread have also been used to estimate the natural structure of forests in an environment of stochastic fire disturbance. LANDIS, for example, is a model of landscape disturbance and succession developed by Mladenoff *et al.* (1996) which employs a cellular method of stochastically modeling fire spread throughout large landscapes. The Boreal Forest Landscape Dynamics Simulator (BFOLDS) is a similar model of landscape change designed specifically for the Boreal forest using a cellular fire spread model based on the CFFDRS (Perera *et al.* 2002). The Sem-Land disturbance and succession model also includes a cellular fire spread model similar to that of BFOLDS, however the fire spread is stochastic rather than deterministic (Li *et al.* 1997).

2.3. Simulation Optimization

A traditional optimization problem can be defined in terms of a set of decision variables x, a feasible region θ and an objective function $F(x)$:

[4] $$\min_{x \in \theta} F(x)$$

The goal of this formulation is to find x^* where $x^* \in \theta$ and $F(x^*) < F(x) \; \forall \; x \in \theta$. In other words, we wish to find a configuration of the decision variables which lies within the feasible region and returns the smallest possible value of the objective function. The feasible region is defined by a set of constraints restricting certain configurations of decision variables. Linear constraints take the form of $Ax < b$. The choice of minimizing $F(x)$ is arbitrary and solving for the maximum value of $F(x)$ is an equivalent problem.

Optimization problems of this form assume that $F(x)$ is a deterministic function and that the closed form is known (i.e. that it can be expressed analytically in terms of "well-known" functions). In other words, for any given solution x, the objective function can be quickly calculated and contains no random elements. Knowing the closed form of the objective function allows for the use of mathematical programming techniques such as linear or non-linear programming which can be used to solve even large optimization problems consisting of thousands of decision variables and constraints (Hillier and Lieberman 2005). Mathematical programming has seen widespread use as a highly successful planning tool in many areas. For certain applications, however, the system being studied may be too complex to derive a closed form function that accurately evaluates a given set of decision variables (Law and Kelton 1991). In these cases it may be possible to use assumptions, simplifications and theoretical models to obtain an

approximate closed form function; but doing so can result in a model that is too simplistic for real world use (Gosavi 2003).

The alternate method to finding a closed form objective function is to *approximate* an objective function using *computer simulation.* Computer simulation allows for a sophisticated method of evaluating the quality of a given solution in a highly complex system (Gosavi 2003). A simulation model can be thought of as a set of *factors* (inputs) and a set of *responses* (outputs); and when a simulation model is presented with a set of factors it will return a set of responses based on the execution of the simulation (see Figure 2)

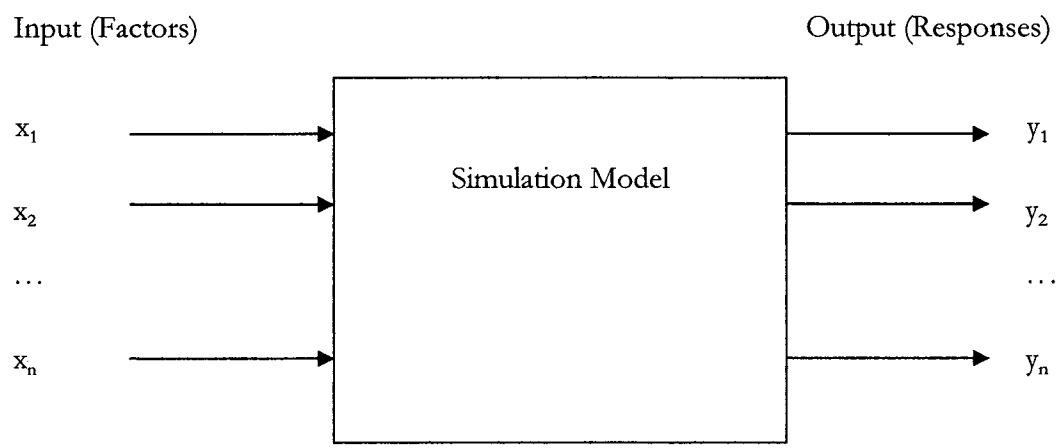

Figure. 2. Inputs and outputs of a simulation model.

When a simulation model is stochastic, the same factors will not necessarily yield the same response but will conform to some probability distribution. The simulation optimization problem is concerned with finding a set of factors that returns an expected optimal response from the simulation program; hence, the stochastic simulation optimization problem can be defined as:

[5] $$\min_{x \in \theta} F(x) = E[\hat{f}(x)]$$

where x is a set of decision variables (which may be thought of as inputs to a simulation program) and θ is the feasible region. Similar to the traditional optimization problem, we wish to minimize the objective function $F(x)$, the difference being that $F(x) = E[\hat{f}(x)]$ where $\hat{f}(x)$ is the *response* of a computer simulation and $E[\cdot]$ is the expectation operator.

Optimization of a stochastic objective function *via* simulation is a difficult problem to solve computationally. This is because the objective function value of each feasible solution cannot be evaluated directly, but must be estimated *via* multiple stochastic simulations. This adds error or *stochastic noise* to each objective function calculation.

Hence the objective function calculated through simulation can be thought of as

[6] $$\hat{f}(x) = f(x) + \varepsilon(x)$$

where $f(x)$ is the *true* value of the objective function and $\varepsilon(x)$ is the error obtained through the stochastic simulation process. Stochastic noise can be diminished and is often nearly eliminated by performing a large number of simulation runs (Law and Kelton 1991). However, since simulation runs are generally computationally expensive an increase in the number of runs performed would reduce the number of alternatives that could be explored in a given period of time (Banks *et al.* 2004). The following section reviews approaches to solving the simulation optimization problem where the goal is to thoroughly search the feasible space while still being able to identify optimal or near-optimal solutions in the presence of stochastic noise.

The use of simulation optimization as a method of solving complex optimization problems has seen a remarkable growth in recent years both in the academic world and in practice.

2.3.1. Solving Simulation Optimization Problems

Approaches to solving simulation optimization problems vary depending on the structure of the feasible region. Solution methods can be divided into those designed for problems where the feasible region is *continuous* and problems where the feasible region is *discrete* (e.g. integer or binary). Continuous valued problems are a class of their own and typically rely on methods such as gradient approximation or response surface methodology (Olafsson and Kim 2002). Since the problem of interest in this study is discrete valued, only solution methods designed for discrete valued problems will be examined in detail.

When using a stochastic simulation model to calculate objective functions, the exact value of the objective function value for any feasible solution can generally not be known exactly. This uncertainty can be regarded as stochastic noise. When stochastic noise is introduced to optimization problems, such as when the objective function is based on a stochastic simulation, a third concern is introduced: *estimation.* Estimation refers to the amount of certainty with which the analyst can infer that one solution is superior to another (Prudius 2007). A higher degree of certainty requires more simulation runs which, in turn, require more computational time. Hence, a major objective in the design of a solution method for simulation optimization is to limit the cost of exploring the search space thoroughly yet still be able to identify high quality solutions in the presence of

stochastic noise. For these reasons, methods of solving deterministic optimization problems are either not applicable or require some modification (Alkhamis *et al.* 1999).

Solution methods for discrete simulation optimization problems can be classified into those designed for problems with small feasible regions and those with sufficiently large feasible regions making complete enumeration unfeasible.

When the number of feasible solutions in a simulation optimization problem is finite and adequately small, such that evaluating all solutions can be completed in a reasonable amount of computational time, ranking and selection (R&S) and multiple comparison procedures (MCPs) can be used.

2.3.2. Ranking and Selection

The concept of R&S was introduced by Bechhofer (1954) when he described the problem of finding the largest mean value from a sample of normal populations. The experimenter collects observations from each population, which are treated as realizations of random variables, and uses these observations to provide a statistical guarantee regarding the quality of the chosen population in relation to the other populations. R&S techniques can be classified as either indifference zone procedures or subset selection procedures. Indifference zone procedures identify an optimal solution but do not discern between two solutions that are sufficiently close, while subset selection procedures are methods of producing a subset from the set of possible solutions that contains the best configuration with a user-defined probability (Swisher and Jacobson 1999). Early R&S techniques (e.g. Paulson, 1964; Bechhofer *et al.* 1968) were of limited practicality to most simulation optimization problems since they required that the variance of the solutions be

equal and known, a condition rarely met with simulation models. Modern R&S techniques have eliminated this requirement (e.g. Dudewicz 1976) and have extended the procedure to multivariate optimization (e.g. Goldsman,1987).

Examples of R&S use in literature include Pichitlamken and Nelson (2003) who used R&S to help guide a random search procedure and Gray and Goldsman (1988) who applied an indifference zone R&S procedure for choosing the best airspace configuration for a European airport. R&S procedures have also been used as a method to "clean up" after simulation optimization; i.e., when an optimization method has identified a number of good solutions with no guarantee as to which is the best, R&S can be used to find the exact optimum (Boesel *et al.* 2003). Several commercial simulation optimization packages employ R&S techniques including AutoStat (Carson 1996) and OptQuest® (Glover *et al.* 1996).

2.3.3. Multiple Comparison Procedures

Multiple Comparison Procedures (MCPs) differ from R&S procedures in that the goal is to *quantify* the difference between two or more solutions rather than selecting the optimal. This difference is analogous to that between hypothesis testing and interval estimation (Yang and Nelson 1991). MCPs use pairwise comparisons and confidence intervals to infer the differences between all proposed solutions, but cannot guarantee a decision of the best solution (Swisher and Jacobson 1999). Recently, efforts have been made to unify both R&S and MCPs to simultaneously run both procedures (e.g. Gupta and Hsu 1984). A unified method would allow an analyst to select an optimal system while

also gaining insight into how much better the solution is in comparison to other possible systems (Swisher and Jacobson 1999).

2.4. Metaheuristic Algorithms

When the feasible region of a discrete optimization problem is infinite or sufficiently large that it would be infeasible to evaluate all possible solutions, the above procedures become useless and instead the analyst must turn to *metaheuristic* procedures that explore the solution space with the objective of finding an optimal or near-optimal solution while only evaluating a fraction of the possible solutions. Metaheuristic algorithms are not concerned with "solving" an optimization problem (i.e. finding an exact optimum), but rather with finding a "good" feasible solution in a fraction of the time required to solve the problem exactly.

Metaheuristic search algorithms explore the solution space using strategies of *diversification* and *intensification* (Blum and Roli 2003). Diversification requires broad exploration of the solution space while intensification involves exploiting specific, promising areas of the solution space. The most effective metaheuristics employ a balance of both strategies (Blum and Roli 2003). The strength of metaheuristics lies in their ability to escape from local optima (diversification), which involves occasionally accepting worse solutions in the course of the search procedure. This ability allows metaheuristics to outperform methods such as greedy heuristics or local search algorithms which only accept new solutions that improve the objective function, causing the search to become trapped in a local optimum which could potentially be significantly worse than a global optimum.

Metaheuristic algorithms can be divided into two major categories: those that use employ a directed search technique and those that are population based. Directed search algorithms are defined by their use of a neighbourhood structure. At each iteration in the optimization procedure there is only one current solution which can move to a neighbouring solution based. In this sense a single solution will travel the solution space in search of a high quality solution. Population based metaheuristics, on the other hand, at all times maintain a set of potential solutions referred to as the current population. At each optimization iteration certain solutions will be removed from the current population and new solutions will be added based on defined rules in the metaheuristic algorithm.

A few of the most commonly used metaheuristic algorithms are: simulated annealing, tabu search, genetic algorithms and scatter search.

2.4.1. Simulated Annealing

Simulated Annealing (SA) is a directed search metaheuristic initially proposed by Kirkpatrick *et al.* (1983) as a search algorithm for deterministic combinatorial optimization problems. The inspiration behind SA stems from the physical act of annealing metals where molecules assume a low energy configuration when cooled with an appropriate schedule. The SA procedure begins with an initial solution and at each iteration either accepts or rejects a neighbouring solution stochastically based on the difference between the objective functions of the two solutions and a temperature variable that decreases with each iteration. The randomness of the rejection or acceptance allows for the search procedure to escape from local optima. SA was adapted as a method of solving simulation problems with stochastic noise by Gelfand and Mitter (1989) and it has

been shown that when the stochastic noise is assumed to be normally distributed, the modified SA methods will converge in probability to a set of local optima (Alkhamis *et al.*, 1999). SA methods have been used in several simulation optimization studies. Examples include Haddock and Mittenthol (1992) who used SA to determine the optimal inputs to a hypothetical factory simulation program and Morito *et al.* (1993) who used SA to find a dispatching schedule to minimize tardiness for a commercial flexible manufacturing system. A simulated annealing algorithm is included in the Witness® simulation software package (Markt 1997).

2.4.2. Tabu Search

Tabu search (TS) is another directed search procedure which differs from simulated annealing in that neighbouring moves are not chosen randomly but instead are the best possible neighbouring solution at each iteration. However, TS is different from a pure hill-climbing technique since worse solutions can be accepted and a memory strategy is employed to prevent cycling back to the same solution in the form of a "tabu list". The tabu list keeps track of recently visited solutions and restricts the search from revisiting or backtracking through the solutions space (Glover and Laguna 1997). The tabu list can be modified in many ways by varying the length of the tabu list, how long solutions are kept in memory, the form of the list etc., and many studies have been focused on finding tabu list configurations that optimize the search. Tabu search has been used to solve many complex optimization problems including job shop scheduling (Nowicki and Smutnicki 1996), the vehicle routing problem (Gendreau *et al.* 2001) and the quadratic assignment problem (Dell'Amico *et al.* 1995).

2.4.3. Genetic Algorithms

Genetic algorithms are the most widely known of the population-based metaheuristics. Rather than having a single current solution at each point in the search, a set of candidate solutions is kept which is likened to a population of biological creatures. The inspiration for genetic algorithms comes from the theory of evolution. The binary strings representing solutions is treated like DNA which may be passed to offspring solutions through combination with other solution DNA. The fitness of each solution is determined by the resulting objective function and at each optimization iteration the weaker solutions will be removed and the fittest solutions will pass on their genetic code. It is hoped that desirable "traits" are passed down to the offspring solutions and that random mutation operations may find improved solutions.

GAs have been employed in a number of theoretical simulation optimization studies (e.g. Tompkins and Azadivar 1995) and are used as search procedures in several commercial simulation optimization packages such as AutoStat (Carson 1996) and RISKOptimizer (Nersesian *et al.* 2001).

2.4.4. Scatter Search

Scatter search is another population-based metaheuristic algorithm first introduced in Glover (1997). It differs from genetic algorithms in that there is no analogy to the survival-of-the-fittest theory where solutions are treated as genetic codes which are combined and mutated. Instead, a set of reference solutions is kept and new solutions are created through weighted linear combinations of these solutions. Rounding mechanisms are employed to ensure that the linear combinations satisfy integer feasibility conditions.

Scatter search algorithms can vary greatly, but are defined by their use of the *scatter search template* (Glover 1998). This template consists of five methods, each of which can be modified to suit a specific problem instance. These methods are:

1) a diversification generation method,
2) a improvement method,
3) a reference set update method,
4) a subset regeneration method, and
5) a solution combination method.

The diversification method is used to spread out the search by generating randomly created solutions which become trial solutions. The improvement method could be any algorithm used to improve a trial solution into an enhanced trial solution, e.g. a simple hill-climbing method. The reference set update method is a way of updating the current set of best solutions, known as the reference set, by replacing a solution in the reference set with a trial solution if it meets some defined criteria. The subset generation method is used to determine at each iteration a subset of the reference set that should be used to create new solutions, and the solution combination method is how these solutions are combined into new solutions.

A good example of the scatter search template can be found in the OptQuest® optimization software package. The general scatter search optimization strategy is as follows. First an initial feasible solution population is created, either randomly or based on user-defined starting points (diversification generation method). The size of the initial population can be adjusted considering the amount of time required for each objective function iteration. After the initial population has been created, the iterative search

procedure begins. At each iteration, two "parent" solutions from the current population are chosen. This choice is influenced by the quality of the solution as well as the memory which stops previously evaluated solutions from being repeated (subset regeneration method). These parent solutions are used to create offspring solutions using linear combination based on the scatter search methodology (solution combination). The worst parent solution is replaced with the best offspring solution and the next iteration begins (reference set update method).

3. METHODS

3.1. Mathematical Formulation of Simulation Optimization Problem

The following formulation is a combinatorial optimization approach to the spatial fuel treatment problem using binary variables to represent the decision of which fuel treatments to implement. It is presented as an abstract and flexible formulation that could easily be tailored to suit different fuel types, fuel treatments, etc. In the case study this formulation will be applied to a real landscape which should give the reader a greater understanding of how the model's parameters and indices can be used. The formulation assumes that the forest is divided into a set of cells, each of which has an associated value, measured in dollars, which would be lost if destroyed by fire. The goal of the formulation is to minimize total fire risk of all cells in the next fire season, where risk equals the probability that each cell will burn multiplied by the value of that cell, summed over the set of all cells.

Indices and sets

$i, I = index\ and\ set\ of\ all\ potential\ fuel\ treatments$
$j, J = index\ and\ set\ of\ all\ cells\ in\ forest$
$N_i = set\ of\ treatment\ prescriptions\ in\ conflict\ with\ treatment\ i$

Parameters

$c_i = cost\ of\ scheduling\ fuel\ treatment\ i$
$v_j = value\ of\ cell\ j$ (\$)
$B = available\ funds\ in\ budget$ (\$)

Decision Variables

$$x_i = \begin{cases} 1 \; if \; treatment \; i \; is \; selected \; for \; implementation \\ 0 \; otherwise \end{cases}$$

Functions

$\hat{f}_j(\boldsymbol{x}) = the \; probability \; of \; cell \; j \; burning \; next \; fire \; season \; given \; treatment \; schedule \; \boldsymbol{x}$
$E[\cdot] - \; Expectation \; operator$
$z = Fire \; risk$

Objective Function

$$Minimize \; z = \sum_{j \in J} E[\hat{f}_j(\boldsymbol{x})] \, v_j \qquad [7]$$

Constraints

$$\sum_{i \in I} x_i c_i < B \qquad [8]$$

$$x_j + \sum_{k \in N_j} x_k \leq 1 \qquad \forall \, j \in J \qquad [9]$$

$$x_i \in \{0,1\} \qquad \forall \, i \in I, j \in J \qquad [10]$$

The objective function, given by equation [7], is to minimize expected fire risk (z) by choosing an optimal set of fuel treatments, represented by $\boldsymbol{x}$. Fire risk is calculated by summing over all cells in a forest the probability the cell will burn multiplied by the value of the cell. The probability that a cell will burn depends on the set of fuel treatments chosen for a particular solution and is represented by the function $\hat{f}_j(\boldsymbol{x})$. This function is calculated through multiple simulations of the stochastic fire simulation model. Equation [8] places a constraint on the total cost of all selected treatments; i.e., the total cost must not exceed the budget parameter B. Equation [9] states that no two conflicting treatments

may be implemented in the same solution. The set of treatments that are in conflict with treatment j is represented by the set N_j which is a user controlled parameter. This could be used, for example, if two treatments covered the same cells but each represented a different type of fuel treatment and the two are mututally exclusive. The final constraint, represented by equation [10], ensures that all decision variables are binary.

The function $\hat{f}_j(\boldsymbol{x})$ could be any type of noisy real valued function which returns the burn probability of cell j based on the set of fuel treatments $\boldsymbol{x}$. A noisy function means that we can never know the true burn probability of cell j; i.e. $\hat{f}_j(\boldsymbol{x}) = f_j(\boldsymbol{x}) + \hat{\varepsilon}(\boldsymbol{x})$ where $\hat{\varepsilon}(\boldsymbol{x})$ is random noise. We define $\hat{f}_j(\boldsymbol{x})$ as the response of a stochastic computer simulation model which estimates burn probabilities based on simulated fire behaviour. The random error $\hat{\varepsilon}(\boldsymbol{x})$ arises from the fact that $\hat{f}_j(\boldsymbol{x})$ are expected values calculated from multiple iterations of a stochastic model, and hence, will not be exactly known.

To solve this optimization problem, two major components are required: 1) a fire simulation model capable of predicting burn probabilities, and 2) an algorithm for solving the optimization problem. In the following section I will describe the structure and design of these two components.

3.2. Design of a Fire Simulation Model (FastFire)

In the previous section we saw that the proposed formulation requires a spatial fire simulation program capable of predicting burn probabilities. Although there are a number of such models (e.g. Prometheus, FARSITE) that are quite accurate and quick to execute, they are typically not well suited for simulation optimization. Simulation optimization

requires that both the optimization model and the simulation model have a well structured interface for easily passing data between them, since this will be done at each iteration of the optimization process. Also, the data output from the simulation program must be compatible with the optimization procedure. Since the above mentioned models were not designed for purposes other than simulation optimization, they are fairly inflexible to change and difficult to incorporate in an optimization process. In order to avoid unnecessary complications, a fire simulation program, FastFire, was designed specifically for use in a fuel management optimization procedure. FastFire was designed with the intent that:

i) It should be well suited to providing a realistic evaluation of burn probabilities,

ii) It should execute quickly (to make simulation optimization feasible),

iii) The data required for the model be readily available for many forests in Canada; and,

iv) Data should easily exchanged with an optimization model.

To achieve these goals, FastFire has the following key components: it is a) based on the Canadian Fire Behaviour Prediction System, b) stochastic and c) cellular. Cells are assumed to be rectangularly shaped and uniform in size. FastFire was not designed to improve upon existing fire simulation models, but to emulate and to modify them with added flexibility.

The core component of all calculations concerning rate of spread in FastFire is the Canadian Forest Fire Behaviour Prediction System (FBP). The FBP is a fire behaviour model that is capable of predicting the rate of spread and intensity of wildfire based on fuel, weather and topographical inputs. The FBP, however, is not capable of predicting

how a fire will progress through a heterogeneous landscape over time (although it does provide an estimate of fire perimeter size in homogeneous fuels as a secondary output). FastFire, in effect, builds a spatial component upon the well established principles of the FBP, which was chosen for its ability to accurately predict fire spread rates in the most commonly found fuel types in Canadian forests. Using the FPB also ensures that fuel data required by the model can be easily obtained for Canadian forests from spatial forest inventory data which can be converted into the sixteen fuel classifications as designated by the FBP.

FastFire is a stochastic model, meaning that ignitions and spread are based on random draws from a probability distribution. Hence, two simulations run with identical initial conditions are unlikely to produce identical results. Stochastic models are commonly used for applications where many fire events are simulated in a large forest area. An excellent example of this is the forest disturbance model the Boreal Forest Landscape Dynamics Simulator (BFOLDS) (Perera *et al.* 2002), a stochastic cellular model used to generate probability maps of forest change over long time periods in forests represented by millions of cells. A stochastic approach to simulating fire spread was chosen for FastFire to ensure that the solutions found in our optimization process are robust enough to protect a forest despite the seemingly random behaviour often exhibited by real forest fires.

FastFire is also a cellular model, meaning that fire progresses through a series of ignitions and extinguishments of regularly shaped areas throughout the forest. The cellular approach was selected for two reasons. Firstly, because our optimization process will require a large number of independent repetitions of the fire simulation procedure, using a

cellular model makes sense since they typically require less computing time than vector based models (Richards 1990; Hargrove *et al.* 2000). Secondly, since our optimization formulation is based on a forest divided into cells and requires that burn probabilities can be calculated for each cell, it makes sense to work with a cellular model. This makes calculating burn probabilities easy since burn probabilities are calculated as the number of times a cell is destroyed by fire divided by the total number of simulation iterations.

3.2.1. Model Inputs

For inputs, FastFire has two requirements: spatial fuel data and weather data. Fuel data are input as a raster dataset which represents the major fuel type, as designated by the FBP, for each cell in the forest. These are simply integer values with each number between one and sixteen corresponding to a fuel type. Cells can also be unburnable land (e.g. rock or clear cut) or water. Weather data is a stream of daily readings of wind direction, wind speed, Fine Fuel Moisture Content (FFMC) and Duff Moisture Content (DMC). FFMC and DMC are indices used by the FBP to predict fire behaviour. These four values are read into FastFire as a stream of data where each line represents the readings for one day. The source of this weather data would typically be from historical weather readings in the area you wish to simulate fire behaviour.

3.2.2. Model Algorithm

FastFire simulates the spread of fire by keeping track of the state of all cells as the model progresses through discrete time steps. Cells can be in one of three states: untouched, burning or extinguished. Fires begin through randomly occurring lightning

strikes that will ignite a cell given appropriate weather conditions. At each time step, this fire may spread to neighbouring cells based on calculations of Rate of Spread (ROS), Flank Rate of Spread (FROS) and Back Rate of Spread (BROS) from the FBP. The model is stochastic and hence having a higher rate of spread implies that there is a greater chance of fire spreading to that cell. Figure 3 illustrates the neighbourhood structure of FastFire.

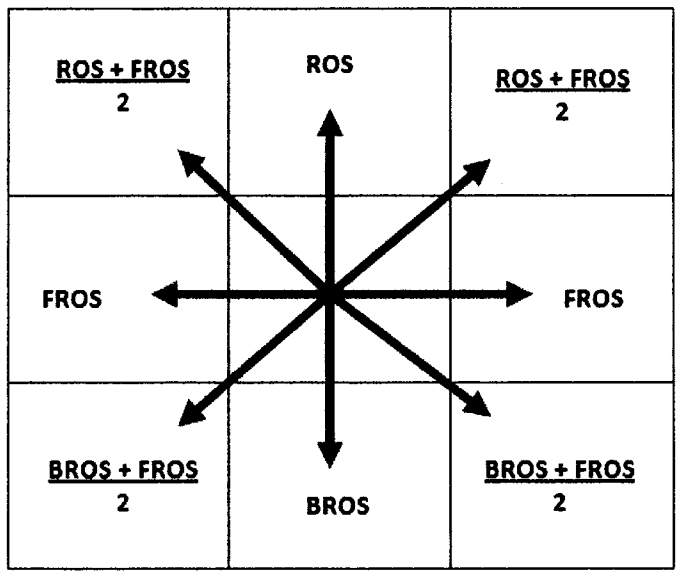

Figure. 3. Neighbourhood structure of FastFire.

Figure 4 gives a high-level overview of how FastFire functions. The idea is that one year of fire activity is simulated and each cell in the landscape that is affected by fire is recorded. To simulate one year of fire activity, first a random number is generated to determine which year of weather data will be used for this simulation. The first day is then simulated by drawing a random number of lightning strikes which could potentially begin new fires. Each lightning strike will affect one cell in the forest drawn randomly and will ignite the cell if weather conditions are conducive (defined as having a DMC > 20). Cells that ignite are then considered actively burning (each of which is stored as a First In/First

Out queue in the code). The day will progress as a number of time steps (defined as a parameter for model calibration), where at each time step burning cells of active fires will randomly spread to neighbouring cells or extinguish. The probability of a burning cell spreading to a neighbouring cell depends on the rate of spread which is calculated as per the FBP which factors in weather and the type of fuel available in the cell. Active fires will continue to burn until they extinguish. At the end of the year long process, every cell that has been destroyed by fire is recorded. This process is repeated for a set number of times (the more iterations, the more accurate the result) and for each cell a burn probability is given as output:

[11] $$Burn\ Probability = \frac{\#\ of\ times\ a\ cell\ is\ affected\ by\ fire}{Total\ \#\ of\ years\ simulated}$$

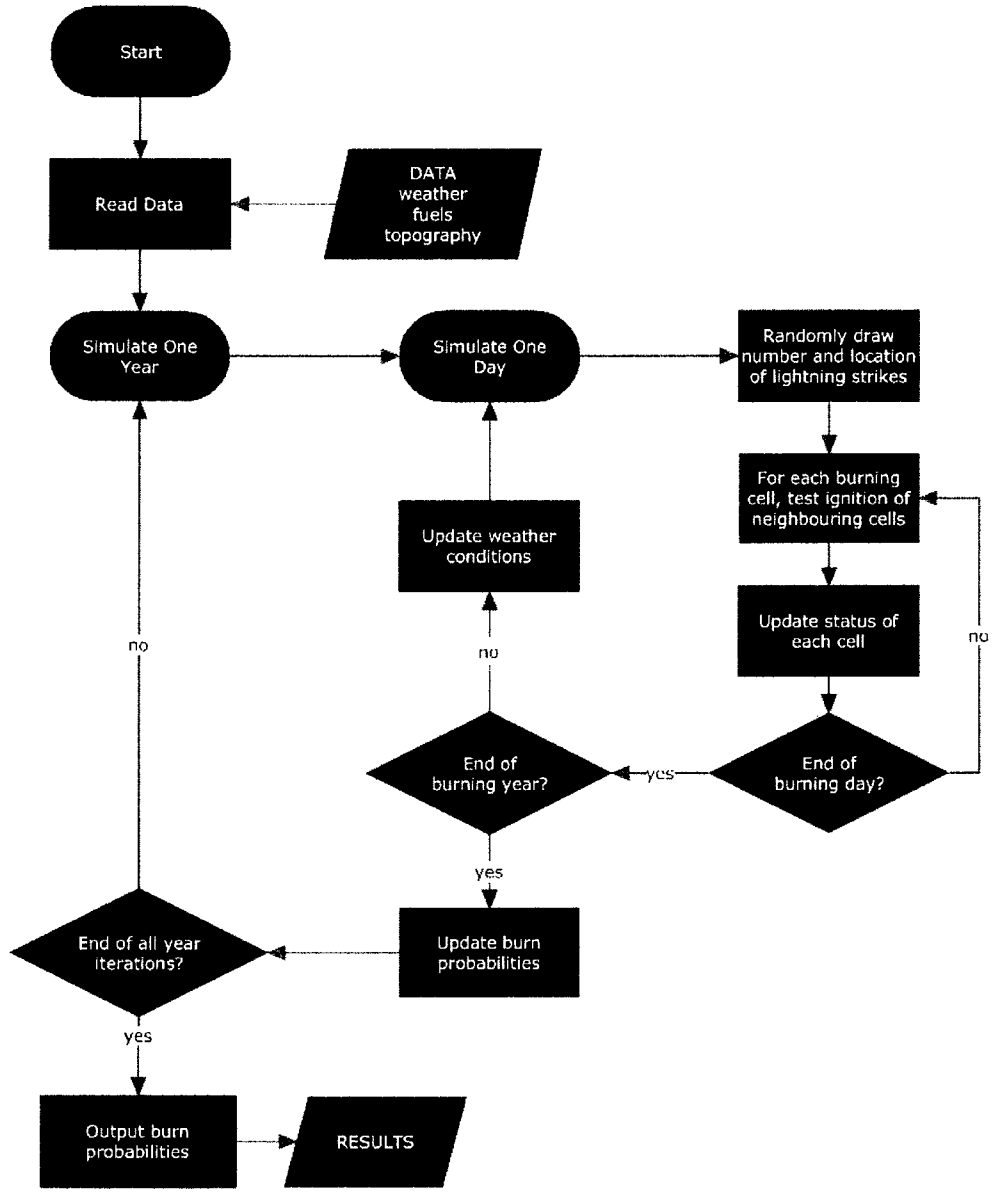

Figure. 4. Flow chart describing FastFire.

3.2.3. Model Parameters

Fire behaviour is controlled in FastFire through the manipulation of three user-defined parameters:

1) The parameter defining the Poisson distribution for the number of daily lightning strikes.

2) The number of time steps a fire has to grow in each day.

3) A scaling parameter for rate of spread.

Poisson rate for lightning occurrence

Fire ignition in FastFire occurs through the random simulation of lightning strikes. Lightning strikes are random both in terms of their occurrence in time and space. Each day in the simulation process where weather conditions meet a required criteria such that ignition is possible (DMC > 20) a random number of lightning strikes will occur throughout the day based on a draw from a Poisson process governed by the variable λ, the rate of the process. A Poisson process is used since it has been shown to best represent the random occurrence of lightning strikes (Wotton *et al.* 2003). The location of the lightning strikes will also be randomized choosing an ignition cell in the landscape based on a draw from a uniform distribution.

The parameter λ when varied will directly affect the average number of fire occurences per year. If $\{N(t), t > 0\}$ is a Poisson process then

[12] $$P[N(t+s) - N(t) = k] = \frac{e^{-\lambda s}(\lambda s)^k}{k!}$$

where λ is the rate of the process and $E[N(s)] = \lambda s$. This equations states that the probability of there being exactly k lightning strikes between time t and time $t + s$ is given by $\frac{e^{-\lambda s}(\lambda s)^k}{k!}$, and from this we can determine that the expected number of lightning strikes is the rate λ multiplied by the length of the time period s.

Number of time steps in a day

Fire advances in FastFire through ignition events that occur at regular time steps. The number of time steps in a day will directly affect the average size of individual fires as

well as average annual area burned. This value will be determined through a calibration process to match model output with historical fire records.

Fire probability scaling parameter

FastFire is a stochastic fire simulation model that simulates fire as a series of probability-driven ignitions between neighbouring cells. The probability of fire advancing to a neighbouring cell is determined by the rate of spread as calculated by the Canadian Fire Behaviour Prediction System. These probabilities are defined as:

[13] $$P(s_{ij}) = \begin{cases} \frac{ROS_{ij}}{\ddot{ROS}} & i \in B, j \in I \\ 0 & else \end{cases}$$

Where:

$P(s_{ij})$ = probability of a spread event occuring between neighbouring cells

ROS_{ij} = rate of spread between cells i and j.

$\ddot{ROS}$ = Maximum rate of spread

B = set of burning cells

I = set of cells with flammable fuels

The final probability can be adjusted using the fire probability scaling parameter φ. Final spread probability was calculated as:

[14] $$\hat{P}(S_{ij}) = \varphi P(s_{ij})$$

Where:

$\hat{P}(S_{ij})$ = Final spread probability

φ = Fire Probability Scaling Parameter.

This parameter will directly affect individual fire size. This parameter will also be determined through model calibration.

3.3. Model Verification and Validation

Once FastFire had been coded it was necessary to test whether the results of the model were realistic. This process followed four procedures described in Sargent (1991) for general computer simulation verification and validation (V&V):

i) Object-Oriented Programming (OOP) for verification,

ii) Bottom up module verification,

iii) Visualization for validation and,

iv) Comparison with historical data for model calibration.

Model verification is defined as the process of ensuring that the written code is a faithful representation of the conceptual model and that no "bugs" exist. Optimally, verification is an exercise that should be continually performed through all stages of code development and not just after the code is complete. During development, verification is simplified by ensuring the code is structured, concise and self-descriptive (Adrion, Branstad and Cherniavsky 1982). At all times, code should be made understandable through good programming techniques and extensive commenting. Object-oriented design and program modularity also ensure the program is well structured and aids in verification since each program module can be individually verified.

Bottom-up module testing involves testing each individual module in the program for correctness first, followed by testing the interaction of modules and finally testing the entire program. This approach is simpler and more effective than simply running the program and trying to trace back where the errors are occurring. Appendix A describes FastFire's component modules, each of which were verified by passing the module inputs checking that the resulting outputs were as expected.

The validity of a model indicates whether its output closely resembles the behaviour of the real world system being modeled. Model calibration was performed using an iterative process and basic validation was accomplished through two tests which combined visualization techniques and comparison with historical data. The first test used visualization; the process of representing output from the simulation model as an image or animation that is easily examined even by those not familiar with the simulation code. This was done by simulating a fire occurrence in completely uniform conditions, i.e. unchanging weather and unvarying fuel structure. Since FastFire is a stochastic model, individual burns tell us little. Instead the average area of 300 burns was spatially plotted. The results were compared with a simple ellipse model for how fire is expected to behave in uniform conditions, and can be found in the results section.

Calibration of FastFire involved comparing historical data with results of the simulation program and modifying the model parameters in an iterative process until they could match defined measures within a given error percent. Although it would be ideal to directly compare burn probabilities of each cell, there is not enough historical data available to accurately estimate burn probabilities at a resolution of one hectare (the resolution used by FastFire). Instead the units of measure used for comparison were: mean area burned per year, mean number of fires over 40 hectares in size and mean individual fire size. The idea here is that although we cannot expect FastFire to predict individual events of a stochastic process with any degree of reliability, we would expect that if the simulation is run for a large number of years in a large forested area that these statistical measures should match within a given error percentage.

To accomplish this goal with FastFire, historical fire data were required for an area as well as historical fuel and weather input data for the same area. Historical Ontario fire data for the period of 1973 to 2006 was obtained from the Ministry of Natural Resource's fire perimeters database. The data consisted of GIS shapefiles spatially representing the areas affected by fire for each year. The attributes for each shapefile contained data for each fire including cause of fire, start date, end date and final fire size. Only fires with a final area greater than 40 ha were recorded in the fire database. This provided us with our historical data. Fuel data and weather data for FastFire were taken from the Ontario Provincial Fuel Database and the Ontario Fire Weather Database respectively.

An area of approximately two million hectares in northern Ontario was used for analysis. The FastFire simulation model was run for 3000 simulated fire years (not sequential year, but the same initail conditions tested for a year 3000 times). where total affected area and individual fire sizes were recorded and the averages were used as a target for model calibration. The process of model calibration is iterative. At each stage the output data are compared with the real data and parameters are adjusted until the output data matches the real data within a desired amount of accuracy, defined arbitrarily as 15%. A summary of the historical data used can be found in Appendix B.

The final test was to graph the number of fires that occurred based on size class. The Forest Management Guide for Natural Disturbance Pattern Emulation (OMNR 2001) demonstrates that, in northern Ontario boreal forest, this graph should take the shape of an "inverse J", meaning that the majority of fires should fall in the smallest fire classes even though the majority of area affected is caused by the largest fires. Graphs were produced

for both historical and simulated data. These graphs showed that 1) the historical data exhibits the expected pattern, and 2) the simulated data closely resembles the historical data.

3.4. Solving the Optimization Problem

Once FastFire was shown to be a reasonable depiction of real fire behaviour for the purposes of this study, it could then be incorporated in our simulation optimization procedure for finding fuel treatment mosaics which minimize fire risk. FastFire comprises the first component require by the optimization problem: a simulation model capable of predicting burn probabilities. The second component was an algorithm for solving the problem.

OptQuest® is a software suite designed to facilitate the development of applications for optimizing complex systems. The engine can be used for any form of linear or non-linear optimization and is specifically designed for use with systems that require simulation optimization.

When optimizing a system represented by a simulation program, the OptQuest® software communicates directly with the user-written simulation model via OptQuest interfaces and classes. The OptQuest solver is flexible and many parameters can be defined by the user to tailor the optimization process to the specific problem. The optimization method follows the scatter search template (see literature review). This general optimization strategy is also augmented by heuristics such as a restarting strategy and an adaptive memory strategy. In the course of the search procedure, the population of solution points may begin to contain many solutions with similar characteristics. This

decrease in population diversity may hinder the search procedure from finding a solution close to the true optimum. In order to counteract this effect, OptQuest may restart the search procedure by creating a new population. This new population is created using a blend of new random points and high-quality "elite" points from previous iterations. The adaptive memory strategy is also a method of increasing search diversity. Since at each iteration of the search procedure, the parent solutions are chosen based on the quality of their calculated objective function, solutions with poor objective evaluations may never be chosen and simply sit in the population pool unused. To counteract this effect, for each iteration that a solution is not chosen as a parent, its attractiveness is increased so that in probability it will eventually be chosen as a parent candidate.

3.4.1 Random and Greedy Heuristics

To determine how well the solutions found by the optimization procedure performed, two other methods of finding solutions to the fuel management problem were examined: randomly chosen solution and greedily chosen solution. The randomly chosen solution method can be explained using the following pseudo code:

$$
\begin{aligned}
&While(total\ cost\ of\ solution\ set\ S < B)\\
&\qquad Randomly\ add\ an\ unchosen\ fuel\ treatment\ to\ solution\ set\ S\\
&End\\
&Return\ solution\ set\ S
\end{aligned}
$$

While the random algorithm has no knowledge of the benefit of the fuel treatments it is choosing, the greedy heuristic does but makes choices with a lack of foresight. All solutions are evaluated by FastFire *individually*, and the choice of which fuel treatments to choose are based on this evaluation alone. The ranking criteria is based on the mean

reduction of fire risd after 10 000 independently simulated fire years for each fuel treatment implented alone.. The important difference between the greedy method and the optimization method is that the greedy solution will have no method of telling how well fuel treatments do in combination with the other treatments that are selected. The greedy solution can be described as:

$$
\begin{aligned}
&For(All\ treatments\ i)\\
&\quad Evaluate\ solution\ i\ using\ FastFire\\
&End\\
&While(total\ cost\ of\ solution\ set\ S < B)\\
&\quad Add\ best\ solution\ based\ on\ evaluation\ to\ solution\ set\ S\\
&End\\
&Return\ solution\ set\ S
\end{aligned}
$$

In summary, the random heuristic will pick a solution with no knowledge of how each treatments perform. The greedy heuristic will choose a solution comprising of the treatments that perform the best individually. The optimization process will a choose a solution by having the knowledge of how all treatments in the solution perform together. It is our hypothesis that this method will be able to find superior methods by exploiting this knowledge.

3.5. Statistical Analysis

Since calculation of a given solution's objective function is based on the results of a stochastic simulation model, one must view objective function values as the realization of a random variable, not as deterministic. The quality of a solution is calculated through repeated simulation, where each simulation returns a burn probability map used to calculate the expected damage caused by fire in one simulated year. After many repetitions of this simulation, a value for fire risk is calculated that equals the probability

of each cell burning multiplied by the value of the cell. The true value, in terms of average fire risk, of a fuel treatment mosaic can never be exactly known. However, statistics can be used to infer that the true value lies within a given region with a given probability; i.e. the objective function value is qualified by a confidence interval.

Law and Kelton (1991) provide a formula for calculating a confidence interval based on repetitions of a stochastic simulation model. This confidence interval is calculated for a given percent confidence α, and uses a t distribution.

[15] $$\bar{X}(n) \pm t_{n-1,1-\alpha/2}\sqrt{\frac{S^2(n)}{n}}$$

In this formula, $\bar{X}(n)$ represents the mean value of a random variable after n independent draws, and $S^2(n)$ represents the variance about this mean. $t_{n-1,1-\alpha/2}$ represents the value returned by the t distribution for n draws when the desired confidence percentage is α. The formula uses two values representing the lower and upper bounds of the interval which will contain the true mean with α confidence. The t distribution is designed specifically for creating confidence intervals of random variable with a normal distribution. Hence, this distribution would not work well if we were to directly use the damage in a landscape after one year of simulation as determined by FastFire, since this distribution does not have normal properties, but has a highly skewed distribution. This skew comes from the fact that there are many years with no damage or very little damage and a much smaller proportion of years with a large amount of damage, giving a left skewed distribution. If we examine the mean fire damage after many simulated years the graph becomes much more normal. Hence equation 15 was used to give an interval for the average value of damage done to a landscape after 1000 simulated years. The central limit

theorem states that the averaged sum of any identically distributed independent random variable will be approximately normally distributed. Hence, by using the summed average of a large number of simulations as our random variable, we can say that the t distibution will give a reasonable estimate of our confidence interval. The use of 1000 simulated years was a somewhat arbitrary choice, but was large enough to give an approximately normal distribution. Hence the confidence intervals are also not exact, but closely approximated.

Confidence intervals for all optimizations were calculated with a confidence percentage of $\alpha = 0.95$. The confidence interval represents the mean fire-risk, calculated as the expected amount of damage done to a forest in a single year. Each confidence interval is based on the average of 1000 simulations repeated 30 times, giving a total of 30 000 simulated fire years for each solution. All simulated years are independent and begin with the same initial conditions.

4. CASE STUDY

4.1. Area Description

Pickle Lake is our study area and is located within the Canadian boreal forest, approximately 530 kilometers northwest of Thunder Bay at the end of Highway 599. The town of Pickle Lake was founded in 1928 as a local transportation centre for mining activities. The area is remote, sparsely populated, and provides raw materials for the logging and mining industries in Ontario. The town of Pickle Lake lies within a formerly glaciated region and the relief is irregular with rocky ridges separating poorly drained depressions and many lakes.

The area supports good tree growth and the development of closed stands on sites where the soil depth is adequate. Black spruce (*Picea mariana (Mill.) B.S.P.*) is the predominant tree, forming stands on thin upland soils, as well as in poorly drained lowlands. Jack Pine (*Pinus banksiana Lam.*) is found on dry sand ridges and rocky slopes of the higher ground. Tamarac (*Larix laricina (Du Roi) K. Koch*) can also be found in the wetter and more open swamps. Occasional mixedwood stands of White Spruce (*Picea glauca (Moench) Voss*) and Balsam Poplar (*Populus balsamifera L.*) are found in river valleys and south-facing slopes.

Pickle Lake experiences long, cold winters with relatively short growing seasons. Average yearly rainfall is nearly 5 cm, most of which occurs in July and August (*Environment Canada* 2008). Occasional hot and dry summer conditions lead to weather conditions resulting in active fire behaviour. Pickle Lake recently completed construction

of a $4.5 million forest fire protection facility in order to combat the increasing threat of forest fire.

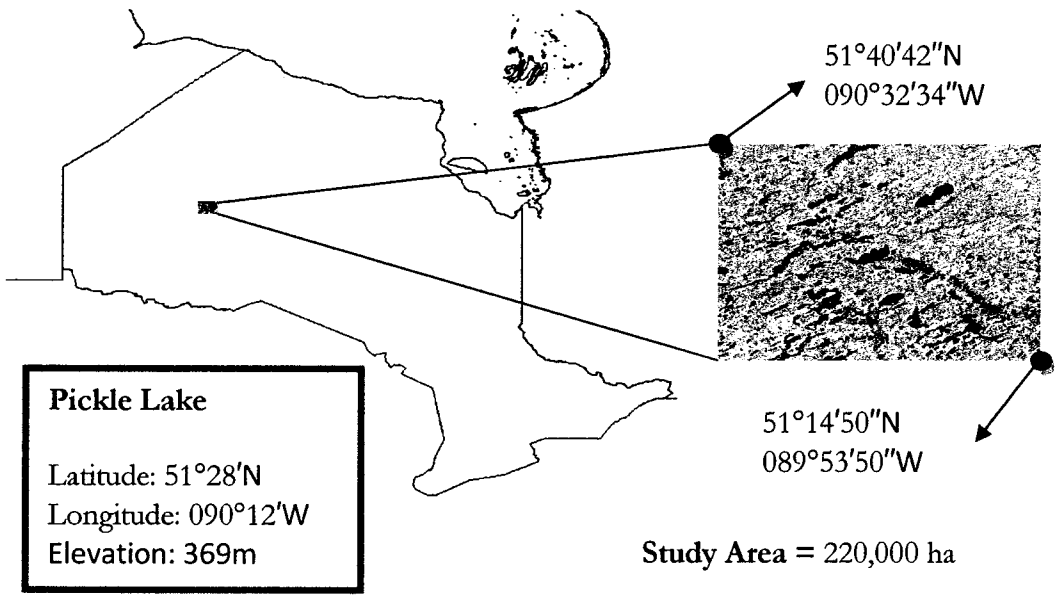

Figure. 5. Case study area.

The area studied represents approximately 220,000 ha of forest surrounding Pickle Lake. Daily weather data from 1960 to 2005 for the area were provided by Ontario's Fire Weather Database. Observations were recorded at the Pickle Lake airport, and represent the daily weather as recorded at noon each day. Fuel data were provided by Ontario's Provincial Fuels Database, a GIS map of fuel composition as represented by the 13 standard fuel types of the Canadian Fire Behaviour Prediction System.

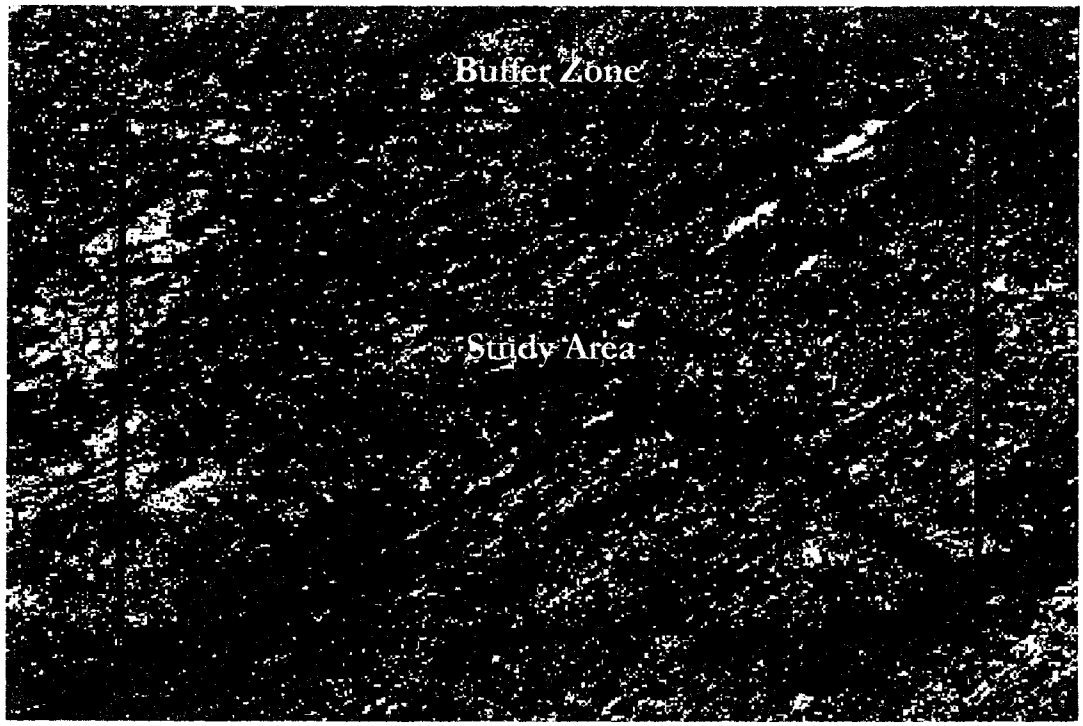

Figure. 6. Total area, comprised of the main study area surrounded by a buffer zone.

The area of the test site was divided into a regularly spaced grid or rectangular cells with each cell of the grid having an area of approximately 1.4 ha. The final size of the grid was 476 by 319 cells, making a total of 151,844 total cells. The area considered for optimization was the central area consisting of 91,106 cells. The outer area was included as a buffer zone to account for fires which could ignite outside the studied region and spread into the center. The main area and buffer zone are shown in Figure 6. Each cell was given a fuel type value based on the 16 classifications of Canadian Forest Fire Behaviour Prediction system. The six fuel types present included Spruce-Lichen Woodland, Boreal Spruce, Mature Jack Pine, Leafless Aspen, Grass and Boreal Mixedwood. Cells represented by water or barren land were set as unburnable. Weather data was imported as a weather stream text file in the form shown in table 1. The weather data used for the study were daily readings from the Pickle Lake airport.

Table 1. Form of input weather stream.

Day	Wind Direction (Azimuth)	Wind Speed (km/h)	Fine Fuel Moisture Content (FFMC)	Duff Moisture Content (DMC)
1	90	4	52.427	1.482
2	180	19	77.689	3.399
3	180	8	25.530	1.342
4	247	24	54.246	1.480
5	292	13	71.484	2.569

4.2. Design of Treatments

One hundred candidate treatments, ranging in size from 60 ha to 400 ha, were manually designed in order to represent a small to moderate sized combinatorial optimization problem. The treatment locations were digitized using desktop GIS software. Each treatment was drawn as a polygon and then transposed onto a raster grid of the same construction as the study area. The only type of fuel treatment considered was clear cutting. Although there are many other kinds of fuel treatments (i.e. thinning, prescribed burns, *etc.*), there are no models that can accurately quantify how these activities affect fire spread. Once a model has been designed that can predict the effects of other treatment activities, they can be easily included into the optimization. The raster was saved as a text file in a format that could be recognized by the fire simulation model, FastFire. Figure 7 visually displays the potential fuel treatments.

The choice of locating the candidate treatments was guided by three principles. The first principle was that areas with high-risk fuels (boreal spruce) in open areas should have options for treatment. The second was that candidate treatment areas should be situated to connect natural spatial features such as lakes when possible. By "linking" lakes it is possible to create treatments that affect a greater area for less cost, an idea was

proposed by Parisien *et al.* (2006). The final guiding principle was that creating a border around the outer edge of the area under consideration could help prevent fires originating outside the area from entering. This idea follows the tactic of "compartmentalization", i.e. sectioning off wooded areas in order to act in a manner analogous to bulkheads on a ship (Loehle 2004). Bulkheads in a ship are designed so that if one section of the ship is breached, the ship will not sink since the water will not spread to other compartments. Similarly, if one area of a forest catches fire we can reduce the total damage done if that fire is contained in a sectioned-off area. For simplicity, the cost of each treatment was uniformly set as $1 per hectare. This not a realistic value, but was used to simplify the problem into constraining total area of all chosen treatments which could be a realistic goal.

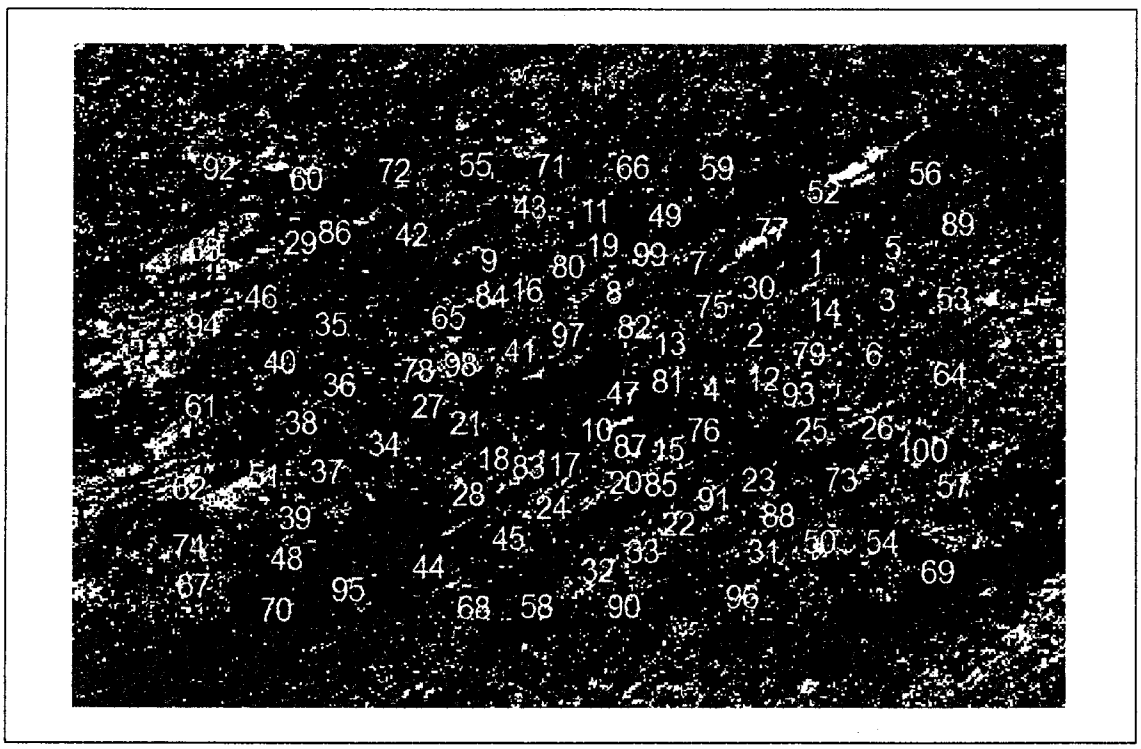

Figure. 7. 100 candidate treatments designed for the optimization procedure.

The value (v_j) of each cell, i.e. the dollar value that would be lost if that cell were destroyed by fire, is directly related to the volume of wood available for harvest in the cell. The area was arbitrarily divided into sections which were each given a random value for m^3 per hectare based on an average age distribution and growth and yield curves for each species as typically found in the boreal forest. This was done since accurate data were not available on the age of the stands in forest, so instead the age classes were simulated to ressemble a real forest. Figure 8 shows the resulting values map, where each section repesents a set of cells with the same m3/ha value. This value is the v_j values in the formulation. Any cells not covered by the value map have a v_j value of zero.

The two other required parameters for the optimization problem were set as follows. The fuel treatment budget was set as $B = \$4\,200$ for all treatments. The N_j values, representing the set of conflicting treatments for a cell, was null for all cells meaning that no treatments were in conflict, and any combination could be chosen. $\hat{f}_j(\boldsymbol{x})$, the function for calculating burn probabilities for each cell, is set as the output of FastFire.

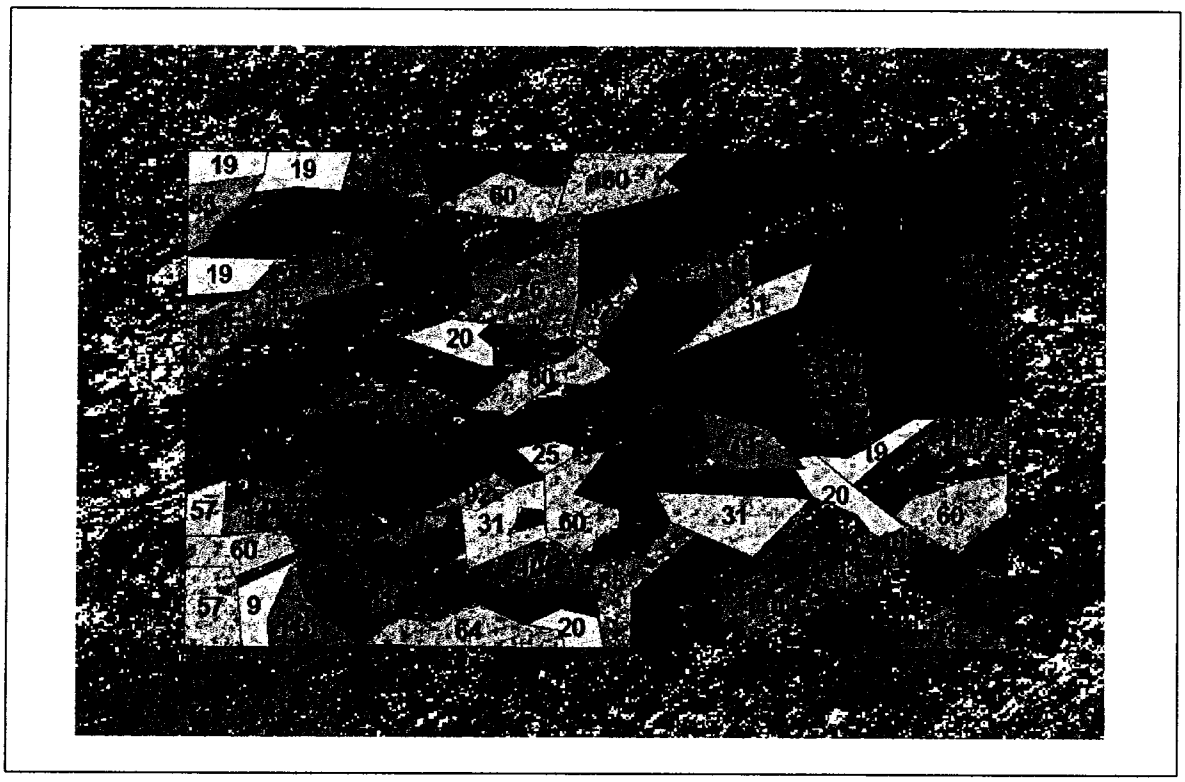

Figure. 8. Values map for the case study area.

4.3. Optimization Software Configuration

The OptQuest® software was configured to represent the optimization formulation described in section 3.1. The decision variables were represented as an array of 100 binary variables. Each decision variable represented one of the potential fuel treatments to be implemented before the next fire season. The cost (in dollars) associated with each treatment was stored in an array of real variables. FastFire was directly integrated with OptQuest® in Java to allow efficient data passing. Each time a year is simulated in FastFire a confidence interval will be calculated in OptQuest® to give an idea of how accurate the estimation of the objective function is. OptQuest® will continue to call

FastFire to perform simulated years (each call will begin in a random year of weather data) until the confidence interval is of a given size with a given percent confidence.

The optimization parameters defining the OptQuest® search were set as follows:

i) For each objective function calculation there will be a minimum of 500 fire years simulated in FastFire and a maximum of 4,000 fire-years simulated.

ii) The confidence interval parameters were set as 3% of the estimated mean with 98% confidence.

iii) The stopping criterion was set so that the optimization procedure will continue until a given number of optimization iterations have been performed.

Nine total optimizations runs were performed with three of these using 2000 iterations, three using 4000 iterations and three using 8000 iterations. All optimizations were run on an Intel® Core™ 2 CPU @ 1.87GHz with 2GB of RAM.

Table 2. Optimization parameters for OptQuest software used in case study.

Optimization Parameter	**Value**
Variables	100 Binary valued variables
Minimum Replications	500
Maximum Replications	4000
Confidence type	Stop replications after minimum replications when confidence level is reached, or when best solution does not fall within current solutions confidence level.
Confidence Level	98%
Percent of mean for which confidence levels are determined	3%
Maximum Iterations	2000, 4000 & 8000
Stopping Criteria	Stop when Maximum Iterations has been reached

5. RESULTS

5.1. Validation of Fire Behaviour in FastFire

Validation and calibration was performed on FastFire using different techniques designed to show that fire behaviour as predicted by the computer simulation model is comparable to recorded fire behaviour. The first test was to see whether simulated fires would take on the expected elliptical shape in uniform fuel and weather conditions. Three hundred independent fire simulations were performed in uniform conditions at three different wind speeds. Each simulation had the same ignition point in on a grid of uniform fuel, set as fuel type Boreal Spruce as defined by the Canadian Fire Behaviour Prediction System. A North West wind was kept at a constant speed. Using the three hundred fire events, three a burn probability map was created for wind speeds of 0 km/h, 15 km/h and 30 km/h. Figure 9 shows the resulting probability maps.

We would expect to see a length-to-breadth ratio that is proportional to wind speed (Alexander 1985). In no wind we found the average fire shape to be approximately circular (a length-to-breadth ratio of 1). In 15 km/h wind the ratio was 1.609 and in 30 km/h wind was 2.13. While this is a fairly rough check, it does confirm that in completely homogeneous conditions (uniform fuel and weather) fire behaves as expected.

Calibration was done to modify FastFire parameters until it could accurately estimate average fire behaviour over a large area. Historical fire data for the time frame of 1973 to 2006 was analyzed for an area of approximately two million hectares of forested land in northern Ontario. From this data three values were calculated:

i) mean annual area affected by fire

ii) mean individual fire size, and

iii) mean annual number of fires

A summary of the historical data analyzed can be found in Appendix B.

The goal was to have FastFire predict all three of these indices within a 15% error margin based on the analysis of 200 simulated fire years. A "guess and check" method was applied where the model parameters of FastFire were varied so that the mean fires per year, mean individual fire size and mean total area burned per year, as calculated by the model, fell within a 15% error margin. After 10 iterations the following parameters were determined:

$\lambda = 2$

$T = 12$

$\varphi = 0.88$

A summary of mean values and percent error for the historical data compared with the fires produced by FastFire is presented in Table 3. We also present a comparison of the standard deviation for each measure in Table 4.

Table 3. Results of adjusting FastFire parameters to match historical behaviour.

	Mean Individual Fire Size	**Mean Number of Fires (>40 ha) per Year**	**Mean Total Area Burned per Year**
Historical Data	9748.01	2.235	21786.82
FastFire Data	9815.65	2.41	23655.73
Error	0.69%	7.83%	8.58%

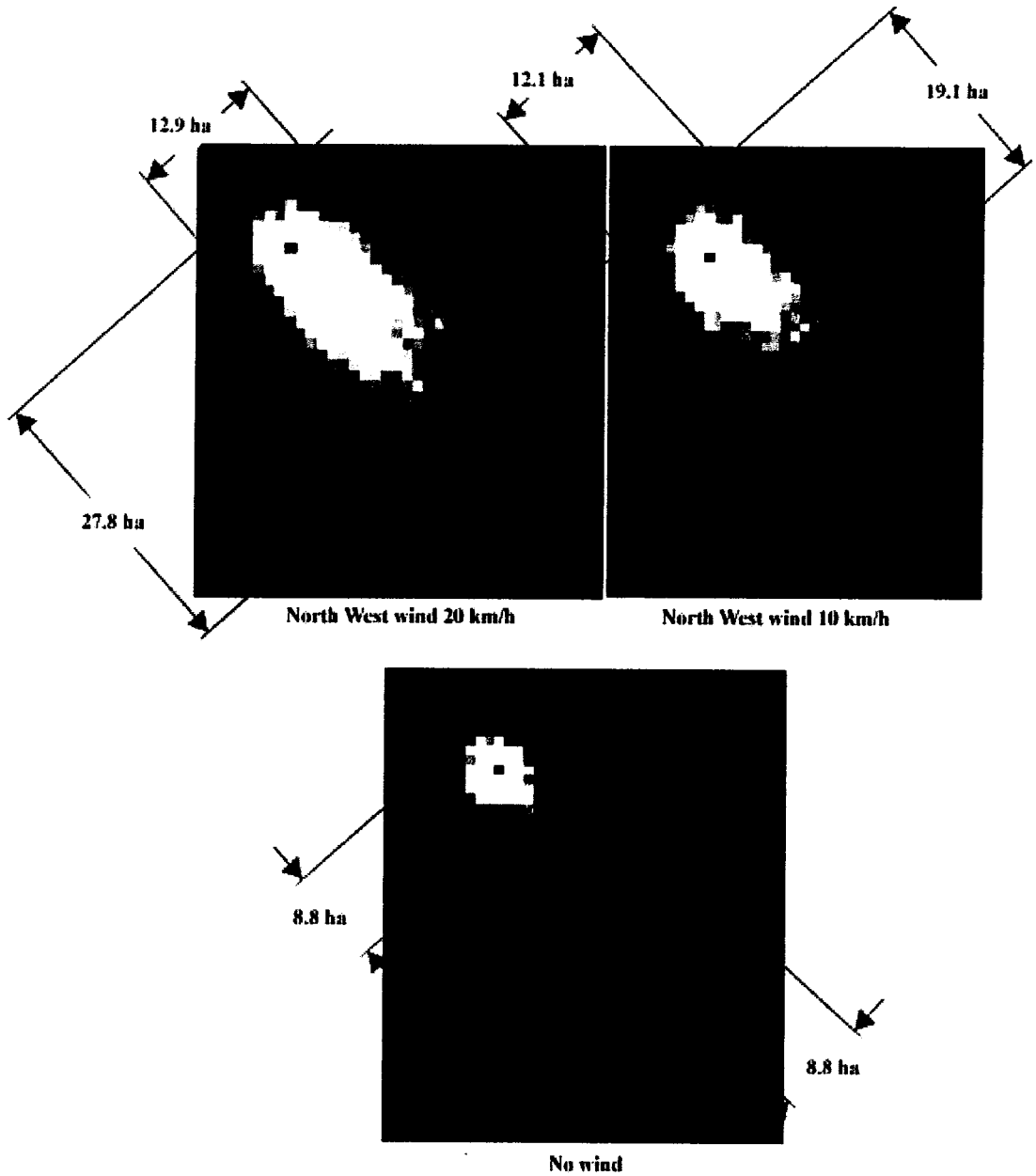

Figure. 9. Fire shapes in homogeneous conditions compared with simple ellipse shapes. The black dot inside fire area represents point of origin (ignition point).

Table 4. Comparing standard deviation of model results with historical data.

	Standard Deviation of Number of Fires (>40 ha) per Year	**Standard Deviation of Total Area Burned per Year**
Historical Data	2.47	16213.87
FastFire Data	3.00	38741.24
Error	21.45%	138.94%

The final test was to graph the number of fires that occurred by size class for both historical and simulated data. This is shown in Figure 10. Blue columns represent the percent of fires falling into each size class from the historic data based on 71 recorded fire shapes (note that these fires did not include any fires with a total area less than 40 ha). The red data represents the same measure based on 200 simulated fires from FastFire.

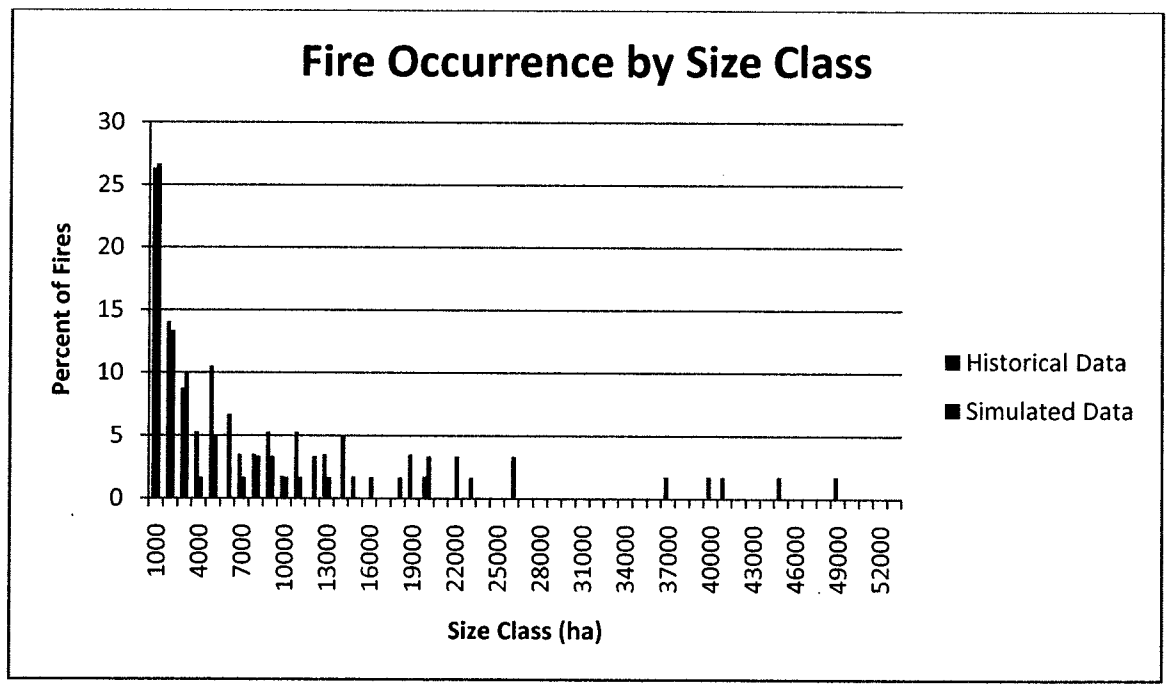

Figure. 10. Fire occurrence by size class for historical and simulated data.

5.2. Simulation Optimization

Three optimizations runs were performed at each 2000, 4000 and 8000 iterations, making a total of nine optimizations. The best solution from each set of three runs was then analyzed by determining a 95% confidence interval for the mean of the objective function (fire risk). Each confidence interval is based on the average of 1000 simulations repeated 30 times, giving a total of 30,000 simulated fire years for each solution. Figure 11 is a graphical representation of the confidence intervals where columns represent mean fire risk and the bars represent the 95% confidence interval as calculated by equation 15. Table 5 summarizes the names used for the various solutions.

The first three columns in Figure 11 represent the solutions found from the set of three optimizations performed with a stopping criterion of 2000 iterations, the second set of three represent those found using 4000 iterations and the following set of three representing those found using 8000 iterations. The column second from the right represents the average fire risk value of randomly generated solutions. The far right column represents the fire risk value for the solution found using the greedy heuristic method. As would be expected there is a noticeable downward trend in fire risk as the number of iterations used in the optimization procedure is increased. However, more iterations did not necessarily ensure a better solution, as is noted with solution 6 having a higher fire risk than solutions 2 or 3 even though it is a solution found with twice the number of optimization iterations. This is not surprising since our optimization method was based on a stochastic metaheuristic. The fire risk of random solutions was significantly higher than all other solutions and also exhibited greater variability. The

greedy heuristic solution was outperformed by all optimization solutions except for one, and was very clost to optimization six.

Table 5. Description of optimization processes performed.

Solution Name	Description	Number of Iterations/repetitions
Optimization 1,2 & 3	Optimizations employing OptQuest and FastFire	2000 iterations
Optimization 4, 5 & 6	Optimizations employing OptQuest and FastFire	4000 iterations
Optimization 7, 8 & 9	Optimizations employing OptQuest and FastFire	8000 iterations
Random	Randomly chosen solutions chosen	30 repetitions
Greedy Heuristic	Solution formed using greedy heuristic	1 repetition

5.2.1. Confidence Intervals for the Difference Between Solutions

Confidence intervals were also used to quantify the expected *difference* between two solutions. Using this method we are able to give a range of how much better one solution has performed over another with 95% probability. If we have two random variable X and Y where $X_1, X_2 \dots X_n$ and $Y_1, Y_2 \dots Y_3$ are independent identically distributed (IID) draws of the random variables, we can create a new random variable $Z = X - Y$ where the IID draws are represented by $(X_1 - Y_1), (X_2 - Y_2) \dots (X_n - Y_n)$. If X and Y represent the average fire risk of two solutions, then Z will represent the difference between them. Creating a confidence interval for Z will give an idea of the expected size of the difference. If the confidence interval (a, b) does not contain 0, then we can say that there is a statistically significant difference between the two solutions.

Three optimization solutions were chosen from the total nine optimizations performed. The best solution found from the nine optimizations (#7) was used to compare

against. Confidence intervals were created to compare how well this best solution performed compared with random solutions, a greedily chosen solution, the best solution after 2000 iterations and the best solution after 4000 iterations. These confidence intervals are shown in figure 12.

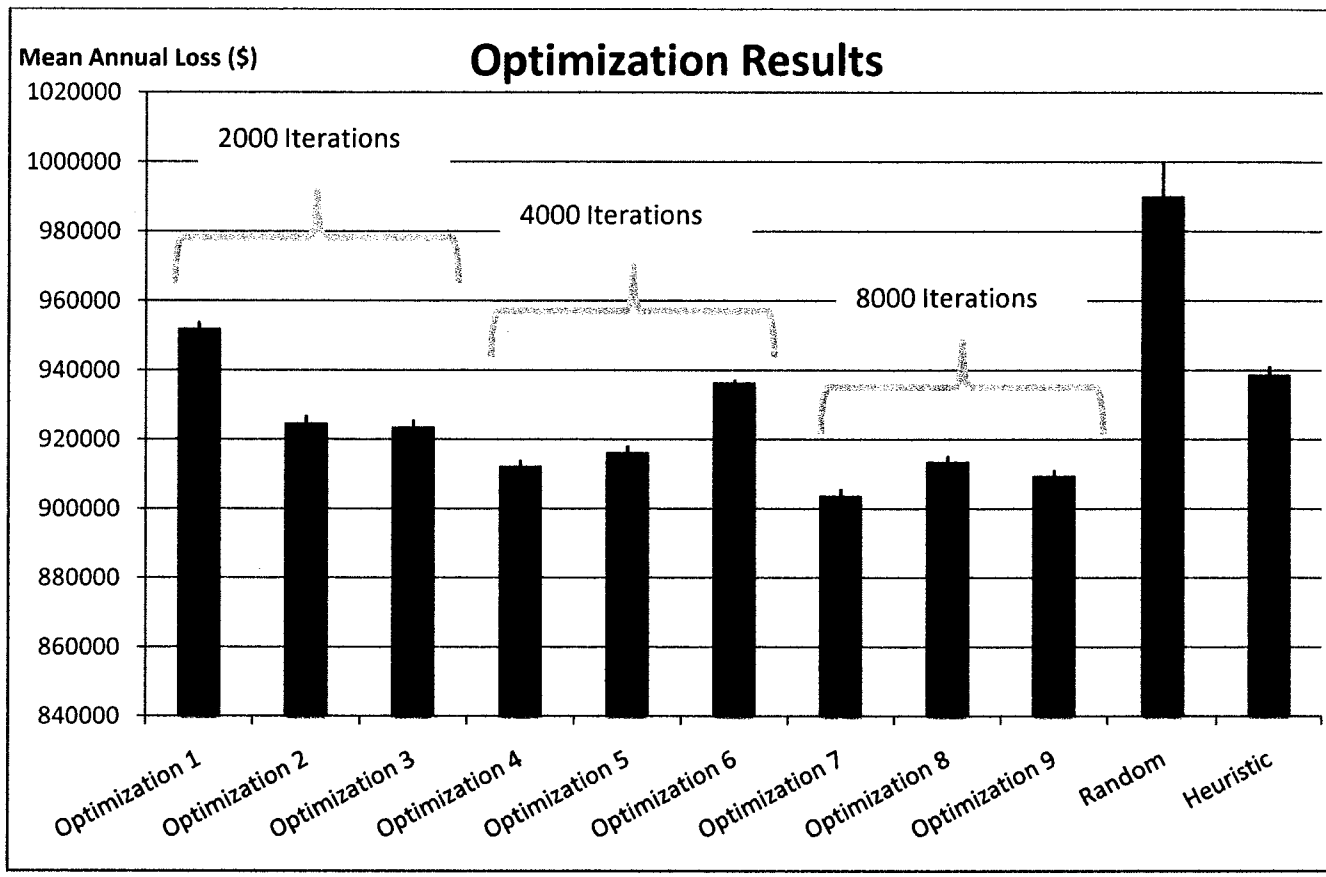

Figure. 11. Mean fire risk (columns) and confidence intervals (vertical bars) for all solutions.

As expected, the best solution did significantly better than solutions that are randomly generated, with a fire risk in the range of 10% lower. The best solution also out-performed the greedy heuristic, but not by as large of margin, having a fire risk approximately 5% lower. It was also shown that the fire risk value for the best solution after 8000 optimization iterations performed better than the solutions found after both

2000 and 4000 iterations. Although the difference is slim, it is statistically significant and this shows that better solutions can be found with more optimization iterations.

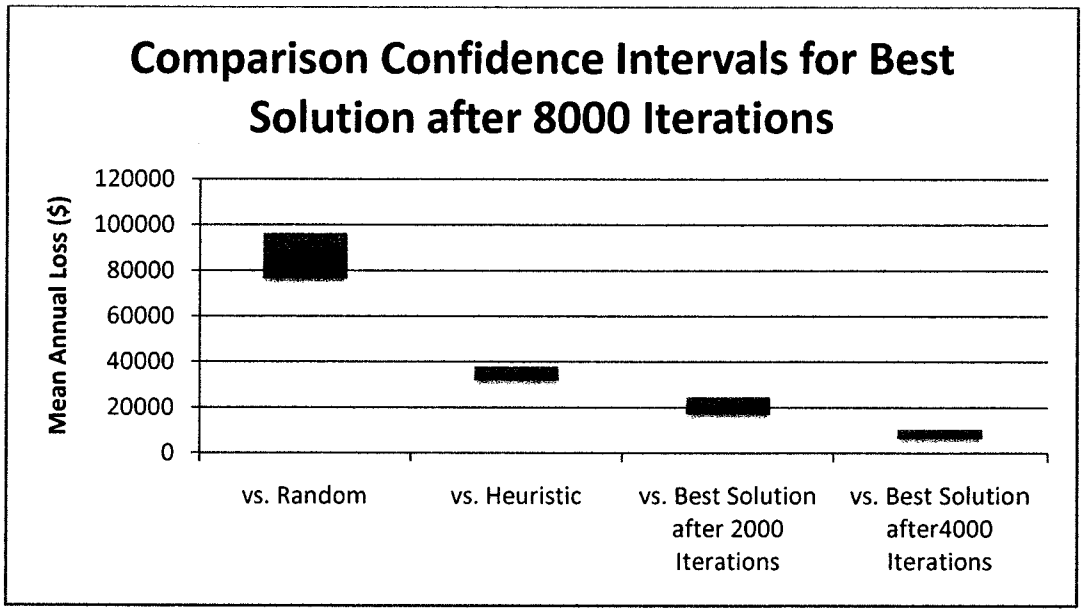

Figure. 12. Confidence intervals comparing solutions.

5.2.2. Optimization Run-Time

The total computational time required to find these solutions ranged from approximately 25 hours to 87 hours depending on the total number of iterations used in the optimization process. As expected, solution quality increased with the number of iterations performed in the optimization procedure. Figure 13 presents a graph of average solution quality compared with the computational time required to find the solution, plotted for all nine of optimizations performed. There is a definite downward trend in the average area affected by fire and it is possible that even better solutions could be found if the search were allowed to continue.

Each optimization iteration took approximately 39 seconds to complete. The length of each optimization iteration was dependent on how many repetitions of the fire simulation program were required to statistically say that with 98% confidence the estimated objective function value for a given solution is within 3% of the true mean. However, OptQuest® also uses a function to determine if a solution is far from the best solution found so far in the search after a user-defined minimum number of simulation repetitions have been performed. If so, it will continue until either the 98% confidence interval or the user-defined maximum number of repetitions has been reached.

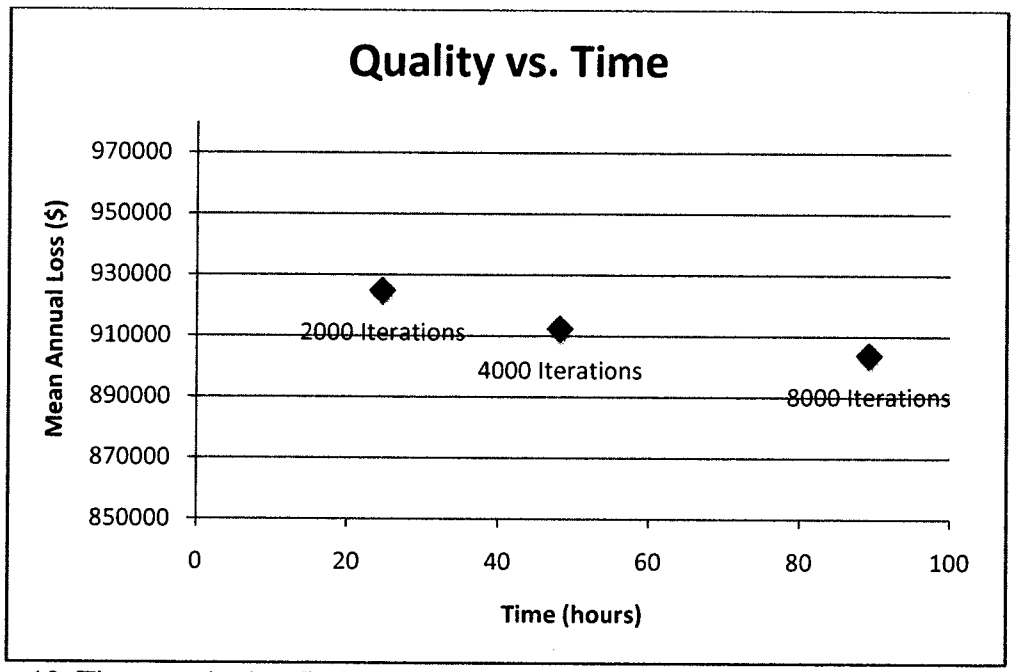

Figure. 13. Time required to find solutions compared against solution quality.

5.2.3. Optimization Progress

The progress of the optimization process for the three optimizations performed with 8000 iterations was graphed to provide some insight into how the solutions were found.

Figure 14 shows the quality of the best solution found at each iteration for the three optimizations.

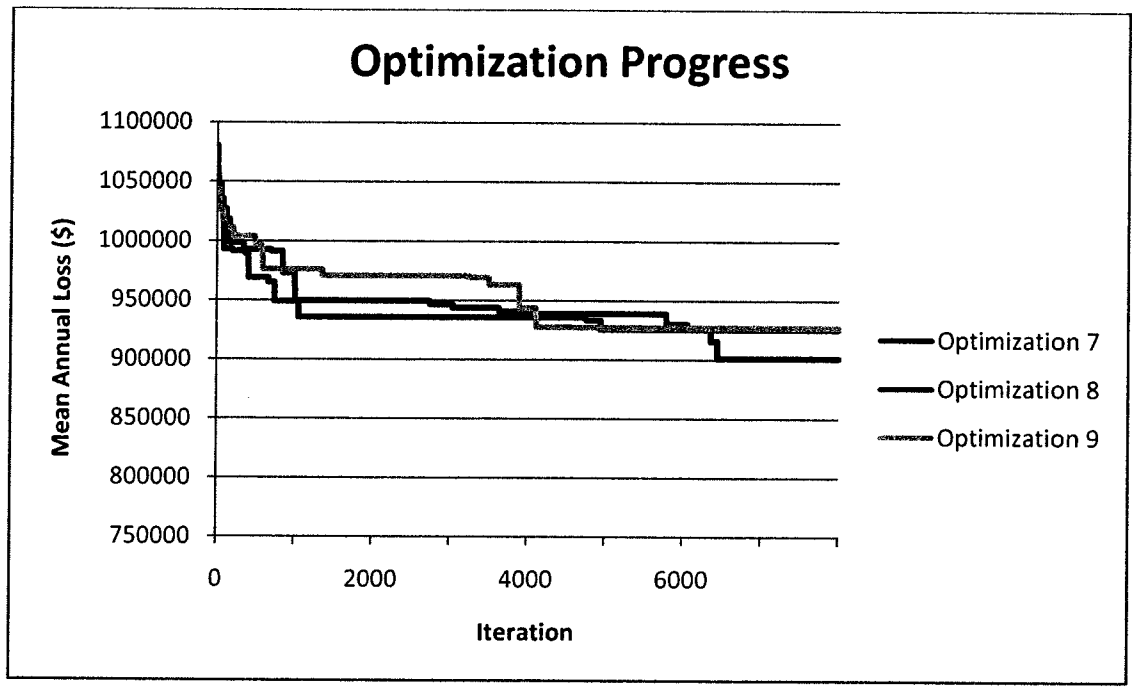

Figure. 14. Optimization progress over 8000 iterations.

5.3. Solution Visualizations

Graphic representations of the solutions were created using GIS software with the goal of examining the spatial layouts of the best solution. ESRI's ArcMap software was used to display a raster file representing a burn probability map generated by FastFire given a specific solution. Shapefiles representing the chosen fuel treatments within the solution were shown overlaying a burn probability map to demonstrate the spatial layout of the treatments and how it affected burn probabilities. The point of this exercise was to determine if any patters were apparent in the solutions that could be generalized as "rule of thumb" recommendations. We present first a burn probability map showing the forested

area without any of the fuel treatments scheduled (Figure 15). The lighter areas represent those cells more affected by fire (on average) than the darker areas. These averages were based on 3000 fire years simulated using Fastfire. The following figures show an example of a random arrangement of fuel treatments (Figure 16), the solution resulting from applying a greedy heuristic (Figure 17), the best solution found after 2000 optimization iterations (Figure 18), the best solution after 4000 iterations (Figure 19) and the best solution after 8000 solutions, which was also the best solution of all (Figure 21). Fuel treatment areas are represented as numbered polygons, where the numbers are simply an identifying ID.

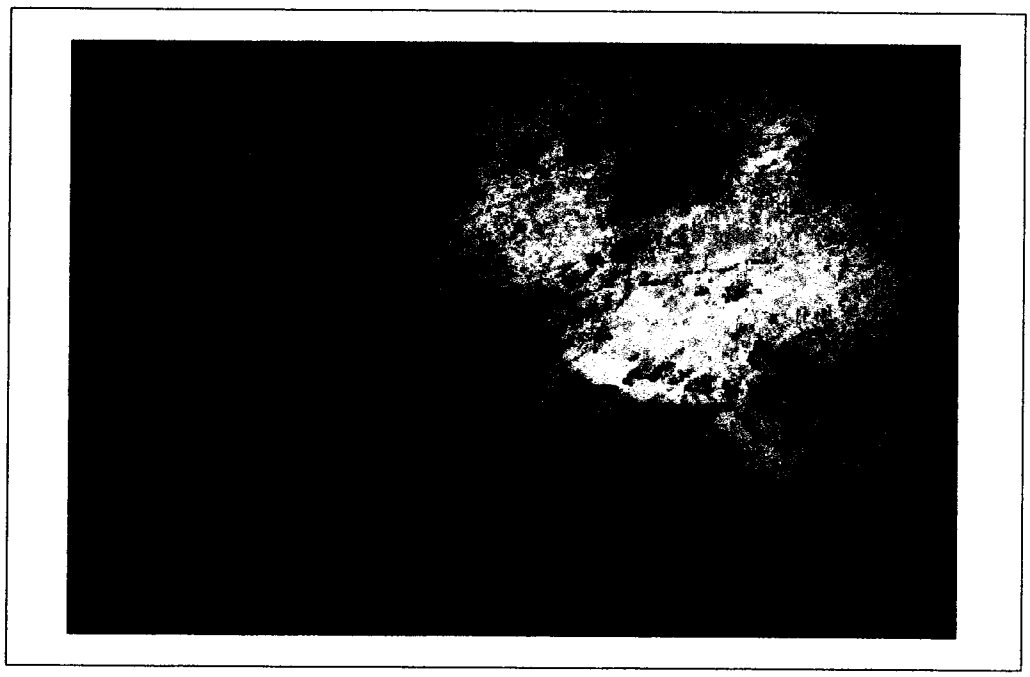

Figure. 15. Burn probability map of untreated landscape. Lighter areas represent higher burn probability.

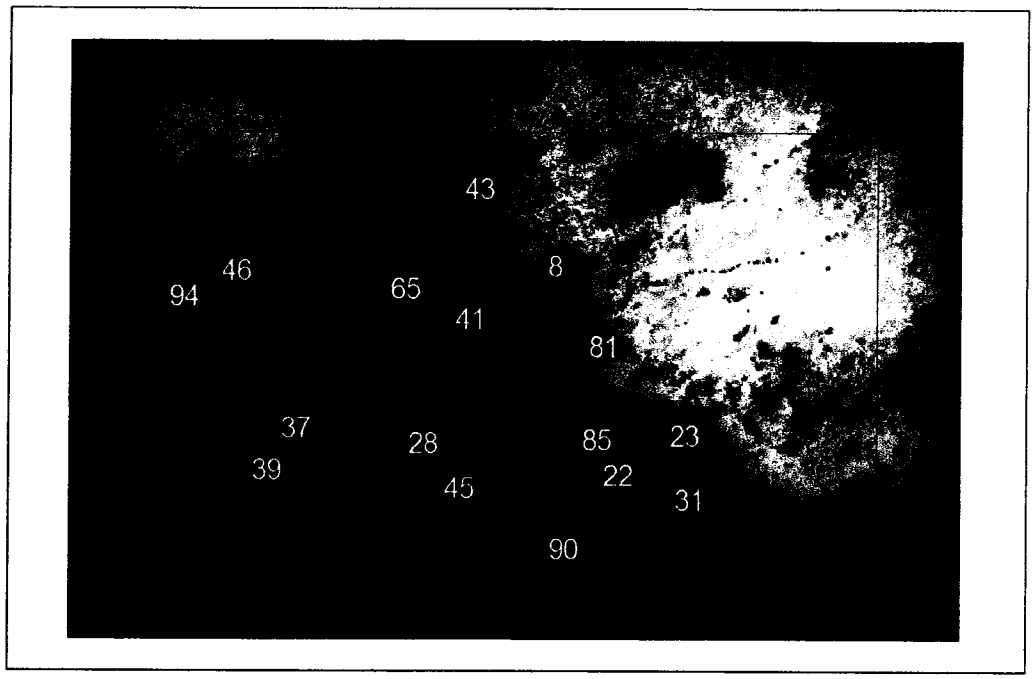

Figure. 16. A random solution with resulting Burn Probability map. Lighter areas represent higher burn probability.

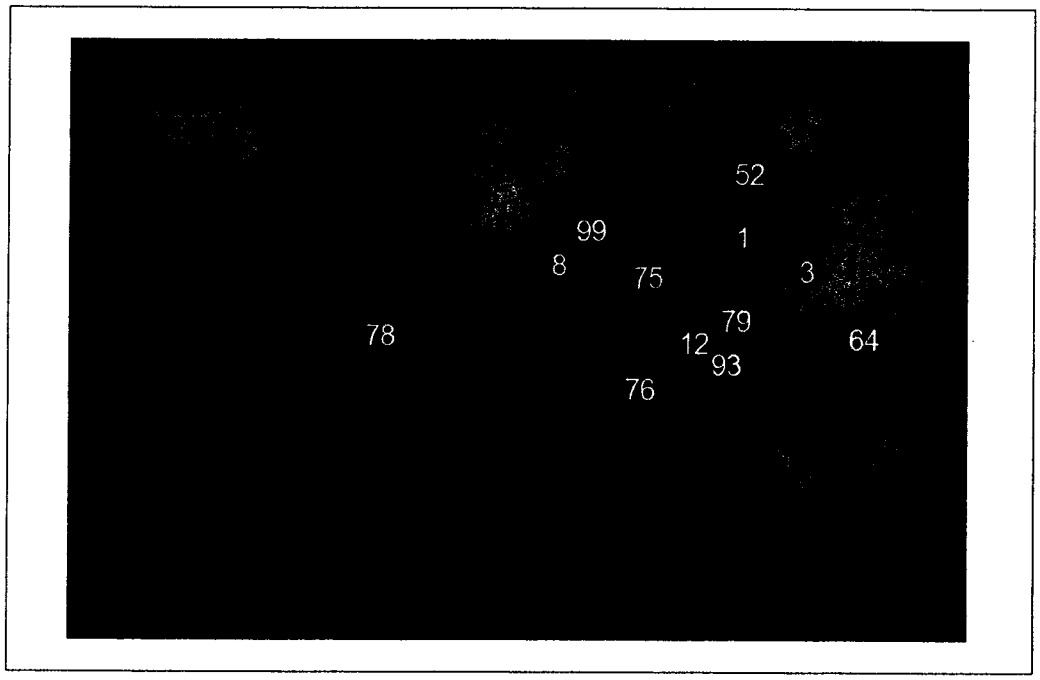

Figure. 17. Greedy heuristic solution with resulting Burn Probability Map. Lighter areas represent higher burn probability.

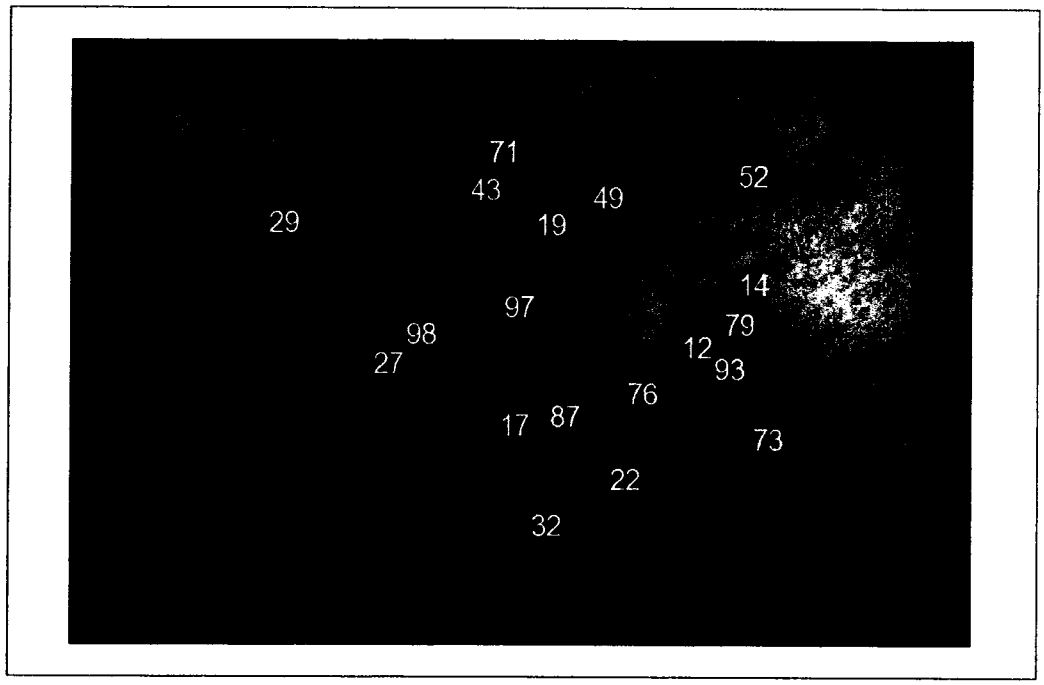

Figure. 18. Best solution after 2000 iterations and resulting Burn Probability Map. Lighter areas represent higher burn probability.

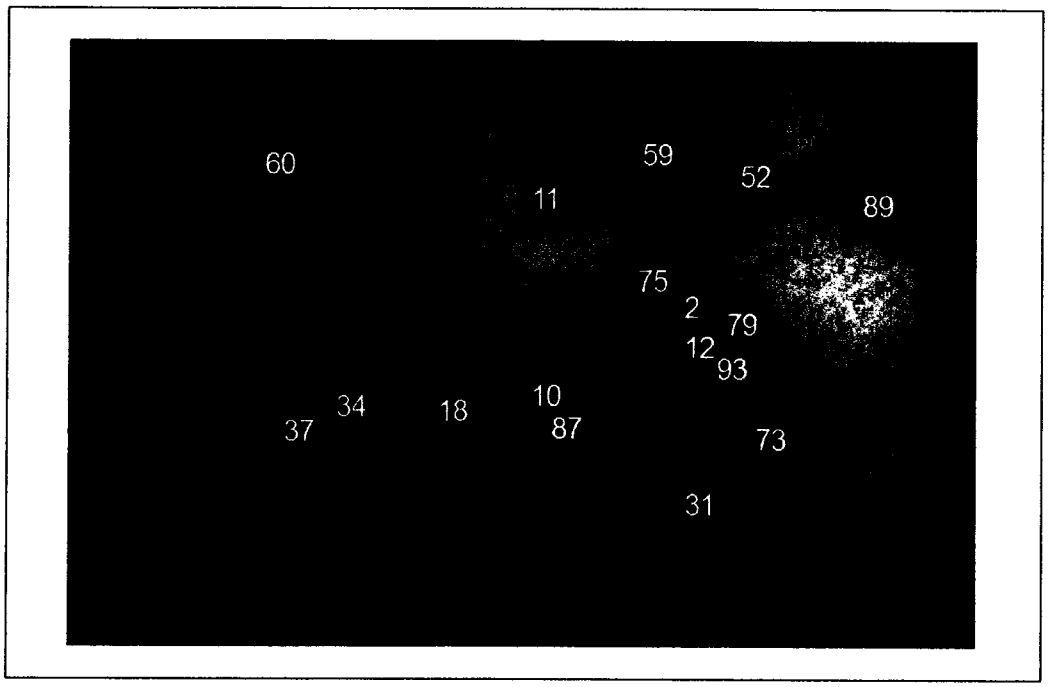

Figure. 19. Best solution after 4000 iterations and resulting Burn Probability Map. Lighter areas represent higher burn probability.

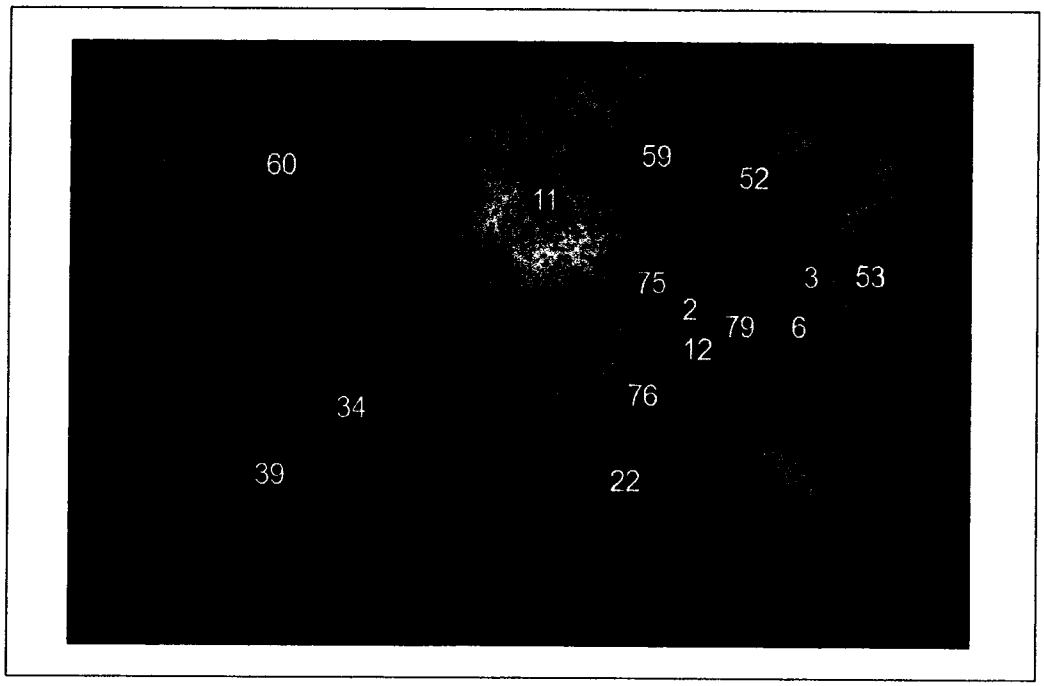

Figure. 20. Best solution after 8000 iterations and resulting Burn Probability Map. Lighter areas represent higher burn probability.

6. DISCUSSION

In this study, simulation optimization was used as a method of locating fuel treatments to minimize fire risk in a forest. The goal was to show that this approach is feasible and that quality solutions can be found using realistic data with moderate sized problems. The discussion will focus on three areas: 1) an evaluation of the strengths and weaknesses of the simulation model, FastFire; 2) an evaluation of the solution-quality produced by the metaheuristic search algorithm; and 3) speculation on the potential usefulness of applying simulation optimization to the harvest scheduling problem.

Computational feasibility is a common road-block in operations research problems. Using simulation based optimization, this problem is greatly magnified since each objective function calculation requires the additional burden of running a simulation program enough times to get a stable response. This idea was kept in mind during the design of FastFire in order to reduce the average time required to simulate one year of fire activity. Wildfire behaviour is complex, and requires a complex numerical representation in order to model with desired spatial accuracy. We could have developed a model that runs in a fraction of the time required by FastFire, but the quality of the simulation, i.e. the resemblance of simulated fires to real fires, would have been greatly affected. Therefore it was a trade-off: speed vs. accuracy. One the one hand, what good is a model that does not reflect realistic fire behaviour? This would provide us no real insight into the fuel treatment problem. On the other hand, a model that takes hours to run would similarly be useless as it would render optimization impossible. Instead, we strove for a middle ground. Any possible method of speeding up the simulation was taken but it was ensured that

FastFire could accurately reproduce historical data. FastFire would not be an ideal model for exactly predicting the spatial progress of a single fire event over time, but the results of our calibration show that its ability to reproduce historical fire behavior of fire on a larger scale is suitable for our purposes.

Several design criteria were used to improve the efficiency of FastFire. Firstly, some fire behaviour characteristics were deemed unnecessary for this study and were not modeled. For example, we did not model fire-branding or distinguish surface fire from crown fire. Fire-branding, or fire-spotting, describes the event when fuels ignite ahead of the main fire front from flying sparks or embers that have been lofted in the wind (Albini 1983). Several fire models, such as EMBYR (Hargrove *et al.* 2000), simulate fire branding as a stochastic event where distance traveled and ignition is generated by a probability distribution. Fire branding has the potential to alter fire behavior; e.g., by lofting over a fuel treatment. Since the behaviour of flying embers is difficult to simulate with accuracy, and the inclusion would be computationally expensive, this aspect of fire behaviour was omitted from FastFire.

Forest fires burn with a wide range of intensities, from smoldering surface fires to complete stand-clearing crown fires. FastFire only simulated the latter; i.e., all fuels in a cell that was affected by fire were considered completely destroyed. Some fire models, such as FARSITE (Finney 1998), will model surface fires separately from crown fires and will also simulate the event of a surface fire transitioning to a crown fire. FastFire, however, is based on the Canadian Forest Fire Behaviour Prediction System which combines both surface and crown fire into a single predictive equation. This approach has

been shown to be accurate for Canadian forests and also reduces the complexity of the simulation model, thereby reducing computational requirements (Stocks 1989).

Despite these departures from some well established models, FastFire was shown to be able to reproduce historical fire data with a desired degree of accuracy for the test region. Calibration was performed on FastFire and it was shown that 1) simulated fires *look* like real fires and 2) the mean and variance of fire size and number of fires per year is close to historical data. As found in our results, the mean value of annual number of fire occurences for simulated fires was within 10% of historical data and the standard deviation was within 25%. The mean annual area burned was also within 10% of historical data, but there was a larger discrepancy in the standard deviation. This discrepancy most likely stems from the fact that the occurrence of very large fires (>30 000 ha) was very rare in the simulated data and occurred more often in the historical data. Plotting fire occurrence by size class showed the expected "inverse J" relationship and closely resembled that of the historical data.

Another way FastFire was designed for fast execution was by optimizing the simulation algorithm. Any shortcuts in the code that could be found were exploited. For example, since weather was input as daily reading, and therefore uniform over each simulated day, rate of spread in each cardinal direction for every fuel type was calculated at the beginning of the day and then read from a lookup-table as required. This small design change reduced the number of calculations required, especially on days with highly active fires. Notwithstanding these efforts, software optimization is not our area of expertise, and there is no doubt that more efficient methods of simulating fire spread could be devised.

In our case study, the time required by FastFire to simulate five hundred years of fire activity was between 34 and 36 to seconds. While this may not seem like a very long time, when the optimization process ran 2000, 4000 or 8000 iterations, and when each optimization iteration required a minimum of 500 simulated years, these seconds start to add up. Hence, if we were able to shave off just one second per 500 simulated fire years, this could save us as much as four hours when running the optimization procedure for 8000 iterations.

Speeding up the simulation model is not the only way to shave time off of the search for optimal fuel treatment mosaics. The optimization algorithm itself can include many techniques for reducing both the number of optimization iterations and the number of simulation repetitions required at each iteration. The goal of any metaheuristic optimization algorithm is to find the best possible solution in the smallest amount of time, and this requires the fewest number of iterations.

With simulation optimization, each candidate solution requires an adequate number of simulations to infer statistically that the estimated objective function value is a good representation of the true value. In generating our results, we required that each objective function be estimated within 3% of the true mean with 98% probability. If we wished to speed up the search, we could have increased the error percentage or decreased the confidence percentage. This, in essence, would reduce the number of required simulation repetitions and could potentially reduce the total time required in the optimization process. Reduced confidence, however, entails a trade-off; for relaxing the confidence interval could result in the optimization returning a large number of *potentially* good solutions

without a statistical support of how good they are. The emergent problem from such a strategy entails determining *which* of these solutions stands out from the rest.

One idea proposed by Boesel *et al.* (2003) is to generate a large population of potentially good solutions and to then pass this population to a Ranking and Selection method which is better suited for finding a single best solution from a smaller set of solutions. In other words, Ranking and Selection could be used to "clean up" after the initial optimization. This approach was not used in this study, but is worthy of examination in future work.

Another feature of OptQuest, which was not used in this study, is the option of incorporating an artificial neural network (ANN) into the optimization process. ANNs are computational models based on biological neural networks which have the ability to "learn" and estimate complex relationships between inputs and outputs. The general idea is that an ANN could be "trained" to predict fire risk by providing inputs (potential solutions) and the resulting output from FastFire (fire risk) during the course of the optimization. After adequate "training", the ANN can predict fire risk with an adequate degree of accuracy in a fraction of the time required by FastFire; i.e., without running the simulation model. Potentially low quality solutions identified by the ANN could then be discarded quickly, rather than being passed to FastFire for a more accurate evaluation. The ANN, in essence, could act as a filter for poor quality solutions so that time is not wasted on fuel treatment layouts that have low likelihood of being high quality, as shown in Figure 21.

In the early phase of optimization, using an ANN would slow the search because of the extra computational time required for training; but, as the optimization progresses, and

there ANN's ability to predict the merit of a proposed solution improves, there is the potential to greatly improve the quality of the candidate solutions that are passed along to the simulation for evaluation of the objective function. This raises the possibility to better solutions in fewer simulation iterations. We chose not to include ANN functionality in our optimization because the complexity of designing and training an ANN was beyond the scope of our objectives.

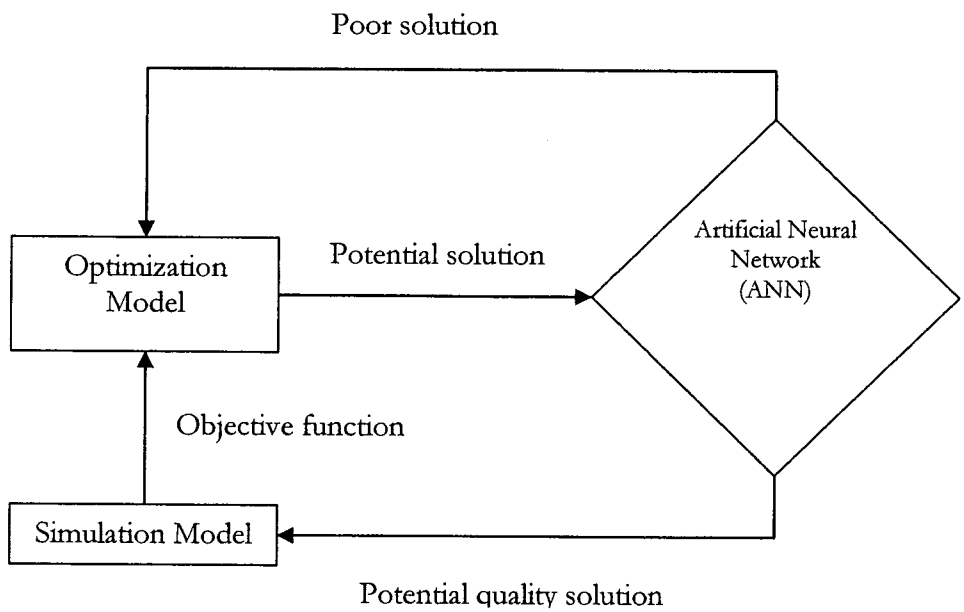

Figure. 21. A trained Artificial Neural Network acting as a "filter" in a simulation optimization process.

The next question to address concerns the quality of solutions found by our method. At the beginning of this study it was imagined that simulation optimization would be a method that could vastly outperform the random and greedy methods of

spatially locating fuel treatments. This belief was driven by the idea that simulation optimization would consider interactions between treatment choices that make up the subtle spatial aspects of the problem that were overlooked by the other two methods. While a greedy heuristic can easily be designed to select treatments that perform well individually, the heuristic has no knowledge of how well all selected treatments perform together, i.e. how their relative spatial layout affects fire spread. The results of this study showed that this insight could be exploited to reduce fire risk. In the following paragraphs we will explore why the simulation optimization approach was able to find better solutions, and why these solutions were not as dramatically better as was expected.

How can we explain that only a 5% decrease in fire risk was noted when comparing the best solution found to a greedy heuristic? Two possible answers to this question are:

1) not enough simulation iterations were performed and the best solution was not close to optimal; or

2) the problem set was not large enough and did provide enough opportunity for the optimization to exploit spatial aspects and the best solution found was near optimal.

We now explore the plausibility of each answer.

The first possible answer implies that the 8,000 iterations performed were not sufficient to find a better, or, near optimal solution. In evaluating this answer, we recall the results illustrated in Figure 14, where the progress made in solution improvements over 8,000 iterations is illustrated graphically. Here we observe that the greatest improvements occurred within the first 1,000 iteration. The final 7,000 iterations reveal

fewer improvements (although there was a notable improvement in Optimization 7 after 6,000 iterations). Hence, it is not unreasonable to conclude that pushing the number of iterations beyond 8,000 would not have improved the solution significantly. Such a conclusion, of course, is qualified by admitting the possibility that the search algorithm was stuck at a local optimum from which it was difficult, to escape.

The second possible reason simulation optimization did not greatly improve over a greedy solution is that the case study used did not provide ample opportunity for improvement; i.e. even the true optimal solution to our problem instance would not be significantly better than the greedy solution. This explanation seems reasonable for several reasons. First, only 100 potential fuel treatment areas were allowed; hence, the problem instance was comprised of only 100 binary decision variables. Although this does provide a large number of potential solutions (estimated as 1.29×10^{27} for our study), this would be considered a small problem in operations research; and intuitively it would seem that the larger the problem size, the greater the potential gain a more sophisticated optimization algorithm has over a greedy heuristic. Second, the case study area itself did not appear to provide a great variation in the landscape with regard to its susceptibility to fire; i.e., by looking at a burn probability map of the area (Figure 15 in Results), we observe that there is only one truly high risk area that stands out.

Perhaps iin a more variable forest with a larger number of potential fuel treatments from which to choose, then the optimal solution would reduce fire risk considerably when compared with a greedily chosen solution. Unfortunately, this brings us full circle to the problem of computational feasibility. A larger problem set requires more optimization iterations which, in turn, requires more time to simulate fire. The

obvious next step in examining simulation optimization as a method of designing fuel treatment layouts is to estimate how problem size relates to computational feasibility.

Quite apart from the solutions' objective function value, solution quality can also be evaluated by examining the mapped solutions (see Figures 16 to 20). The mapped solutions of the greedy heuristic focused exclusively on areas of high fire risk; but the solutions of the simulation-optimization model were more dispersed spatially; and this dispersal reveals a strategy of "boxing off" the high risk areas; thereby restricting fires from both entering and leaving these high-risk areas. The best solution, found after both 8,000 iterations and 4,000 iterations, appears to make a perimeter around the highest risk area, which can be noted in the upper right portion of the forest in Figures 19 and 20. This is an idea that has been mentioned previously by Loehle (2004), who theorized that "compartmentalizing" a forest with fire breaks might be an effective strategy for reducing fire risk. Another common spatial trend of the simulation optimization solutions appeared to be the "use" of natural features such as lakes; i.e., by choosing treatments adjacent to a lake (or treatments on either side of a lake) it extend the natural fire barrier, thus providing greater fire protection for less cost. This is exhibited in the both Figure 19 and 20, which include treatments areas 52 and 59 which are located on either side of a lake. Also, treatment area 60, which forms a link between two lakes, was selected in both the best solution after 4000 iterations and the best solution after 8000 iterations.

A final question to discuss is whether the model developed in this research could be applied to other spatial planning problems in forestry. Of particular interest is the problem of scheduling harvests in an environment of stochastic fire disturbance. We address this question in two parts: first, we examine whether extending simulation optimization to

spatial harvest-scheduling would be of any innovative significance. Second, we address the question of whether it might be feasible.

Regarding the innovation of extending our model to harvest-scheduling, we note that the problem of scheduling harvests in an environment of fire disturbance has already been modeled by researchers using diverse approaches. For example Reed and Errico (1986) first modeled this problem using linear programming; later, Boychuk and Martel (1996) modeled the problem using stochastic linear programming model; and more recently, Armstrong (2004) developed a framework in which a linear programming model interacts with a Monte Carlo burn simulation model. It should be noted that, in all of the preceding work on modeling the relationship between fire and forest harvesting, both the harvest schedules and the fires were modeled aspatially. This is, in part because the questions addressed were of a strategic nature; namely, determining a sustainable allowable annual cut in an environment of stochastic fire disturbance; but it is also because a spatially explicit solution requires binary decision variables, and the computing burden of solving binary integer programming models typically increases exponentially with problem size. One exception to this aspatial trend has been the work of Peter and Nelson (2005), who developed a framework in which a spatially explicit harvest scheduling model interacts with a stochastic burn model. Of note, though, is that the harvest schedules Peter and Nelson developed were entirely deterministic; hence, the cut-block layout was not made with foresight into reducing the risk of fire damage in a managed forest. Hence, should we be able to extend simulation optimization to the spatial explicit, tactical planning problem, the innovation would be authentic; i.e., researchers have not yet developed a spatially explicit harvest scheduling model that optimizes, through a

stochastic fire simulation model, the layout of a spatial pattern of cut-blocks in order to reduce the risk of fire to the managed forest.

This brings us to the second question; *viz.*, whether such an extension is feasible? In effect, we are asking whether: (a) the allocation of fuel management treatments, which has (b) been scheduled over one period, could be transformed into (c) a cut-block allocation model that is (d) scheduled over multiple periods.

Let us first address the initial requirement for such a transformation; *viz.*, the transformation of a model for the allocation of fuel treatments into a meaningful model for the allocation of cut blocks. First, such a transformation must assume that cut-blocks function in a manner analogous to fuel treatments; *i.e.*, that harvesting a stand has a similar effect on a forest's fire risk as removing the fuel from that stand. Such an assumption does not seem problematic, for both models remove the stand's fuel (although the accumulation of slash can affect fire behaviour). Second, a tactical harvest-scheduling model is typically constrained by limits on the total volume of a given stand type (e.g., species, age-class) that may be harvested. In like manner, the fuel treatment allocation model is constrained by a budget, and this budget could easily be transformed into a constraint on total volume of each stand type for which "treatment" is allocated. Finally, the typical harvest-scheduling model has an objective to maximize net present value, while the objective function of the fuel treatment allocation model has the objective to minimize fire risk. Hence, a transformation of the fuel management model into a harvest-scheduling model would require that fire risk be treated as one of multiple objectives or as a constraint. Some experimentation in model formulation would be required; but we would begin by evaluating fire risk only in solutions that are found within the feasible

region defined by constraints. In fact, the more heavily constrained the problem is, the smaller is this feasible region, and the more effective would be our computationally costly search and evaluation procedure.

The second major requirement for transforming a fuel-treatment allocation model into a tactical harvest-scheduling model is the incorporation of a multiple period planning horizon. We can discern two approaches to evaluating fire risk resulting from a schedule of harvests over multiple periods: i) to evaluate all periods simultaneously; or ii) to evaluate the cumulative risk resulting from each period's harvest-layout as time moves forward, i.e., in the manner of a rolling planning horizon. In the first approach, a candidate harvest scheduling solution, specifying which blocks are cut in which period, is evaluated for fire risk as though all cut blocks occurred in one period. The advantage of this is that blocks can be selected in period t because of their ability to reduce fire risk in combination with blocks selected in period t + 1. Hence, a type of foresight is possible in a period by period movement toward reduced fire risk. The disadvantage of this is that it might take many periods for the full effectiveness of the resulting reduction in fire risk to be achieved. In the second approach, a myopic, but immediately optimal achievement of reduced fire risk is achieved in each period planned. One simply reschedules, and re-evaluates fire risk at the beginning of each period, for one period at a time. Hence, in theory, it is possible to extend simulation-optimization to the harvest-scheduling problem; but we must first endeavour to build such a model, and examine the computational challenges, before concluding that it is possible in practice.

7. CONCLUSION

The goal of this study was to design and evaluate a method of spatially designating fuel treatment areas that will reduce the probability of fire damage in a given forest. To achieve this goal, the problem of spatial fuel treatment allocation was formulated as a simulation optimization problem with integer decision variables representing a set of potential treatment areas. To calculate fire risk, a spatial fire simulation model was designed and validated for use in the boreal forest of northern Ontario.

It was determined with 95% certainty that the best solution found using our simulation optimization technique could reduce fire risk by 5% compared with a greedily chosen solution. Computational time required for 8000 iterations of the optimization procedure was approximately 86 hours. Through examining the spatial pattern of selected treatments, we found the best solution employed strategies to minimize fire risk; *viz.*, "boxing in" the high risk zones and the inclusion of natural fire barriers in the arrangement of fuel treatments. The results therefore suggest that simulation optimization has the potential to create intelligently planned fuel treatment mosaics that can reduce fire risk in a forest.

Future research on this problem would involve applying simulation optimization to the spatial harvest scheduling problem. Feasibility of such an application would require discovering methods of increasing the efficiency of the search procedure so that high quality solutions can be found in reasonable computing time.

BIBLIOGRAPHY

Adrion, W. R., M. A. Branstad, J. C. Cherniavsky. 1982. Validation, verification, and testing of computer software. ACM Comp. Surv. (CSUR). 14: 159-192.

Agee, J. K., B. Bahro, M. A. Finney, P. N. Omi, D. B. Sapsis, C. N. Skinner, J. W. van Wagtendonk, C. P. Weatherspoon. 2000. The use of shaded fuelbreaks in landscape fire management. For. Ecol. Manage. 127: 55-64.

Agee, J. K., and C. Skinner. 2005. Basic principles of forest fuel reduction treatments. For. Ecol. Maange. 211: 83-96.

Albini, F. A. 1976. Estimating wildfire behavior and effects. General Technical Report INT-30. Ogden, UT: U.S. Forest Service.

Albini, F. A. 1983. Potential spotting distance from wind-driven surface fires. USDA Forest Service Research Paper INT-309. 27 p.

Alexander, M.E. 1985. Estimating the length-to-breadth ratio of elliptical forest fire patterns. Proceedings of the 8th Conference on Fire and Forest Meteorology. Society of American Foresters, Bethesda, Md. Publ. 85-04. pp. 287–304.

Alexander, M. E., R. A. Lanoville. 2004. The international crown fire modeling experiment fuel treatment trials. Tall Timbers Fire Ecology Conference Proceedings 22: 222.

Alkhamis, T. M., M. A. Ahmed, and V. K. Tuan. 1999. Simulated annealing for discrete optimization with estimation. Euro. J. of Oper. Res. 116: 530-544.

Anderson, H. E. 1983. Predicting wind-driven wildland fire size and shape. USDA Forest Service Research Paper INT-305.

Andrews, P. L. 1986. BEHAVE: fire behavior prediction and fuel modeling system-BURN subsystem, part 1. General Technical Report INT-194. Ogden, UT: U.S. Department of Agriculture, Forest Service, Intermountain Research Station. 130 p.

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

April, J., F. Glover, J. P. Kelly. 2003, Manuel Laguna, Simulation-based optimization: practical introduction to simulation optimization, Proceedings of the 35th conference on Winter simulation: driving innovation, December 07-10, New Orleans, Louisiana

Arca, B., P. Duce , G. Pellizzaro, M. Laconi, M. Salis , and D.Spano. 2006. Evaluation of FARSITE simulator in Mediterranean shrubland. For. Ecol. Maange. 234: 110-115.

Armstrong, G.W. and S. G. Cumming. 2003. Estimating the cost of land base changes due to wildfire using shadow prices. Forest Sci. 49: 719–730.

Armstrong, G.W. 2004. Sustainability of timber supply considering the risk of fire. Forest Sci. 50: 626-639.

Azadivar, F., J. Shu, and M. Ahmad. 1996. Simulation optimization in strategic location of semi-finished products in a pull-type production system, in Proceedings of the 1996 Winter Simulation Conference (Charnes, J. M., Morrice, D. J., Brunner, D. T., and Swain, J. J., eds.), pp. 1123–1128, Institute of Electrical and Electronics Engineering, Piscataway, New Jersey.

Ball, G. L. and P. D. Guertin. 1992. Improved fire growth modeling. International Journal of Wildlife Fire, 2: 47-54.

Banks, J. A., J. S. Carson, B. L. Nelson, and D. M. Nicol. 2004. Discrete-Event System Simulation. Upper Saddle River, NJ: Prentice Hall, 4th ed.

Bechhofer, R.E. 1954. A single-sample multiple decision procedure for ranking means of normal populations with known variances. Annals of Mathematical Statistics 25:16-39.

Bechhofer, R.E., J. Kiefer, and M. Sobel. 1968. Sequential Identification and Ranking Procedures. The University of Chicago Press, Chicago, Illinois.

Bergeron Y., S. Gauthier, V. Kafka, P. Lefort, and D. Lesieur. 2001. Natural fire frequency for the eastern Canadian boreal forest: consequences for sustainable forestry. Canadian Journal of Forest Research. 31: 384-391.

Blum, C., and A. Roli. 2003. Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison. . ACM Computing Surveys. 35: 268–308.

Boesel, J., B. L. Nelson, and S.Kim. 2003. Using Ranking and Selection to "Clean Up" after Simulation Optimization. Operations Research. 51: 814-825.

Burgan, R. E., R. C. Rothermel. 1984. BEHAVE: Fire behavior prediction and fuel modelling system. USDA Forestry Service General Technical Report INT-167.

Canadian Forest Service. 2007. Forest Fire Facts and Questions. Retrieved May 13, 2008 from Natural Resources Canada http://fire.cfs.nrcan.gc.ca/facts_e.php

Carey, H., M. Schumann. 2003. Modifying wildfire behavior–the effectiveness of fuel treatments: the status of our knowledge. Southwest Region Working Paper 2. Santa Fe, NM: Forest Trust, National Community Forestry Center. 26 p.

Carson, J. S. 1996. AutoStatTM : Output statistical analysis for AutoModTM users. In Proceedings of the 1996 Winter Simulation Conference, J. M. Charnes, D. J. Morrice, D. T. Brunner, and J. J. Swain, Eds. IEEE, Piscataway, N.J., 492–499.

Catchpole, T., N. de Mestre. 1986. Physical Models for a spreading line-fire. Australian Forestry. 49: 102-11.

Chandler, C.C., P. Cheney, P. Thomas, L. Trabaud and D. Williams. 1983. Fire in forestry. Volume 1: Forest Fire Behavior and Effects. New York: Wiley.

Conard, S.G., T. Hartzell, M. W. Hilbruner, and G. T. Zimmerman. 2001. Changing fuel management strategies—the challenge of meeting new information and analysis needs. International Journal of Wildland Fire 10: 267-275.

CWFGM Project Steering Committee. 2006. Prometheus User Manual. Unpublished Manuscript.

Dallyn, C., L. Fremont and K. Johnson. 2000. A guide to managing community wildfire risk. Retrieved March 2, 2008, from http://www.saskforestcentre.ca/uploaded/Guide to Managing Community Wildfire Risk.pdf.

Dell'Amico, M., M. Labbe, F. Maffioli. 1999. Exact solution of the SONET ring loading problem. Operations Research Letters. 25: 119-129.

DeSorcey, G. 2001. Chisholm Fire Review Committee - Final Report. Alberta Sustainable Resource Development. LWF-063.

Dudewicz, E.J. 1976. Statistics in simulation: How to design for selecting the best alternative. Proceedings of the 1976 Winter Simulation Conference, 67-71.

Environment Canada. 2008. Canadian climate normals and averages 1971-2000 retrieved June 14 2008, from http://www.climate.weatheroffice.ec.gc.ca/

Fernandes, P., and H. Botelho. 2003. A review of prescribed burning effectiveness in fire hazard reduction. Int. J. Wildland Fire. 12: 117–128.

Filmon, G. 2004. Firestorm 2003 provincial review. Province of British Columbia, Victoria, British Columbia.

Finney, M. A. 1998. FARSITE: Fire Area Simulator—model development and evaluation. Res. Pap. RMRS-RP-4. Fort Collins, CO: U.S. Department of Agriculture, Forest Service, Rocky Mountain Research Station. 47 p.

Finney, M. A. 2001. Design of regular landscape fuel treatment patterns for modifying fire growth and behavior. Forest Science. 47:219–228.

Finney, M.A. and J. D. Cohen. 2003. Expectation and evaluation of fuel management objectives. In: USDA Forest Service Proceedings, RMRS-P 29. pp. 353–366.

Finney, M. A. 2005. The challenge of quantitative risk analysis for wildland fire. For. Ecol. Maange. 211: 97-108.

Flannigan, M. D., K. A. Logan, B. D. Amiro, W. R. Skinner, and B. J. Stocks. 2005. Future area burned in Canada. Clim. Change. 72: 1–16.

Fons, W. T. 1946. Analysis of fire spread in light forest fuels. Journal of Agricultural Research. 72: 93-121.

French, I. A., D. H. Anderson, and E. A. Catchpole. 1990. Graphical Simulation of Bushfire Spread. Mathematical Computer Modelling. 13: 67-71.

Fulé, P.Z., C. McHugh, T. A. Heinlein, and W. W. Covington. 2001. Potential fire behavior is reduced following forest restoration treatments. Ponderosa Pine Ecosystems Restoration and Conservation: Steps Toward Stewardship Proceeding (eds G.K. Vance, W.W. Covington and C.B. Edminster), pp. 28–35. US Department of Agriculture Forest Service RMRS-P-22. Rocky Mountain Research Station, Ogden, UT.

Gardner, R. H., W. H. Romme, and M. G. Turner. 1999. Predicting forest fire effects at landscape scales. Pages 163–185 in D. J. Mladenoff, and W. L. Baker, editors. Spatial modeling of forest landscapes: approaches and applications. Cambridge University Press, Cambridge, UK.

Gelfand, S. B. and S. K. Mitter. 1989. Simulatied annealing with noisy or imprecise energy measurements. J. Optim. Theory Appl., 62: 49-62.

Gendreau, M., G. Laporte, and F. Semet, F. 2001 A dynamic model and parallel tabu search algorithm for real-time ambulance relocation. Parallel Computing 27: 1641–1653

Glover, F., J. P. Kelly and M. Laguna. 1996. New Advances and Applications of Combining Simulation and Optimization. Proceedings of the 1996 Winter Simulation Conference, J. M. Charnes, D. J. Morrice, D. T. Brunner, and J. J. Swain (Eds.), p.144-152.

Glover, F. and M. Laguna 1997. Tabu Search. Dluwer Academic Publishers, Boston.

Glover, F. 1977. Heuristics for Integer Programming Using Surrogate Constraints. Decision Sciences, 8: 156-166.

Glover, F. 1998. A template for scatter search and path relinking. In: Hao, J. K., Lutton, E., Ronald, E., Schoenarue, M., Snyers, D. (Eds.), Artificial Evoluion, Lecture Notes in Computer Science 1363. Springer, pp. 13-54.

Goldsman, D. 1987. Ranking and selection tutorial: 2- factor normal means procedures. Proceedings of the 1987 Winter Simulation Conference, 52-57.

Gosavi, A. 2003. Simulation-Based Optimization : Parametric Optimization Techniques and Reinforcement Learning. Operations Research/Computer Science Interfaces Series. Springer.

Gould, J.S. 1988. Validation of the Rothermel model and related fuel parameters in grassland fuels. In Cheney, N.P. and Gill, A.M., editors, Proceedings of the conference on bushfire modelling and fire danger rating systems, Canberra, 11-12 July, 1988, Canberra: CSIRO, 51-64.

Graham, R.T., S. McCaffrey, and T. B. Jain. 2004. Science basis for changing forest structure to modify wildfire behavior and severity. USDA Forest Service General Technical Report, RMRS-120, Rocky Mountain Research Station, Ogden, Utah, 43 p.

Gray, D. and D. Goldsman. 1988. Indifference-zone selection procedures for choosing the best airspace configuration. Proceedings of the 1988 Winter Simulation Conference, 445-450.

Green, L. R. 1977. Fuelbreaks and other fuel modifications for wildland fire control. Agric. Handb. 499. Washington, DC: U.S. Department of Agriculture. 79 pp.

Green, D. G. 1983. Shapes of simulated fires in discrete fuels. Ecological Modelling. 20: 21-32.

Green, D.G., A.M. Gill, and I.R. Noble. 1983. Fire shapes and the adequacy of fire-spread models. Ecol. Mod. 20:33-45.

Gupta, S.S. and J. C. Hsu. 1984. A computer package for ranking, selection, and multiple comparisons with the best. Proceedings of the 1984 Winter Simulation Conference, 251-257.

Haddock, J., J.Mittenhal. 1992. Simulation optimization using simulated annealing. Computers and Industrial Engineering. 20: 387-395.

Haige, J.W. and K. N. Paige. 2001. Learning SIMUL8: The Complete Guide. Plain Vu: Bellingham, WA.

Hargrove W.W., R. H. Gardner, M. G. Turner, W. H. Romme, and D. G. Despain. 2000. Simulating fire patterns in heterogeneous landscapes, Ecological Modelling, 135: 243-263.

He, H.S., D. J. Mladenoff, and T. R. Crow. 1999. Object-oriented design of LANDIS, a spatially explicit and stochastic forest landscape model. Ecol. Modell. 119: 1–19.

He, H. S., B. Z. Shang, T. R. Crow, and E. J. Gustafson. 2004. Shifley, S. R. Simulating forest fuel and fire risk dynamics across landscapes—LANDIS fuel module design. Ecol. Model., *180*, 135– 151

Hill, S. D. and M. C. Fu. 1994. Simulation optimization via simultaneous perturbation stochastic approximation, in Proceedings of the 1994 Winter Simulation Conference (Tew, J. D., Manivannan, S., Sadowski, D. A., and Seila, A. F., eds.), pp. 1461–1464, Institute of Electrical and Electronics Engineeres, Piscataway, New Jersey.

Hillier, F. S., and G. J. Lieberman. 2005. Introduction to operations research (8th ed.). McGraw-Hill Science/Engineering/Math.

Hirsch, K.G. 1989. Documenting wildfire behavior: an example from the 1988 Brereton Lake Fire, Manitoba. Fire Manage. Notes. 50: 45-48.

Hirsch, K.G. 1996. Canadian Forest Fire Behavior Prediction (FBP) System: User's Guide. Nat. Resour. Can. Can. For. Serv. North. For. Cent., Edmonton, Alta. Spec. Rep. 7.

Hirsch, K., V. Kafka, and B. Todd. 2004. Using forest management techniques to alter forest fuels and reduce wildfire size: an exploratory analysis. In R. T. Engstrom, K. E. M. Galley and W. J. de Groot (Eds.). Proceedings of the 22nd Tall Timbers Fire Ecology Conference: Fire in Temperate, Boreal, and Montane Ecosystems (Pages 175-184). Tallahassee, FL: Tall Timbers Research Station.

Hof, J. and P. Omi, 2003. Scheduling removals for fuels management. In: USDA Forest Service Proceedings, RMRS-P 29. pp. 367–378.

Hong, L. J. and B. L. Nelson. (2006). Discrete optimization via simulation using COMPASS. Operations Research, 54: 115–129.

Keane, R. E., G. J. Cary, I. D. Davies, M. D. Flannigan, R. H. Gardner, S. Lavorel, J. M. Lenihan, C. Li, and T. S. Rupp. 2004. A classification of landscape fire succession models: spatial simualtions of fire and vegetation dynamics. Ecological Modelling. 179:3-27.

Kirkpatrick, S., C. D. Gelatt, and M. P.Vecchi. 1983. Optimization by simulated annealing. Science. 13: 671–680.

Law, A. M. and W. D. Kelton. 1991. Simulation modeling and analysis. McGraw-Hill, Inc., second edition, 1991.

Li, C., M. Ter-Mikaelian, and A. Perera. 1997. Temporal fire disturbance patterns on a forest landscape. Ecol. Model. 99: 137–150.

Loehle, C. 2004. Applying landscape principles to fire hazard reduction. For. Ecol. Maange. 198 : 261-267.

Lynch, D.L., K. Mackes,, 2003. Costs for reducing fuels in Colorado forest restoration projects. In: USDA Forest Service Proceedings, RMRS-P 29. pp. 167–175.

Malanson, G. and L. Trabaud. 1988. Computer simulation of fire behaviour in garrigue in southern France. Applied Geograph., 8: 53-64.

Mangan, R. 1999. Wildland fire fatalities in the United States: 1990–1998. Technical Report 9951-2808-MTDC. US Department of Agriculture, Forest Service, Missoula, MT, p. 14.

Markt, P. L. 1997. Witness simulation software: A flexible suite of simulation tools. Retrieved July 8, 2008 from http://csdl2.computer.org/persagen/DLAbsToc.jsp?resourcePath=/dl/proceedings/&toc=comp/proceedings/wsc/1997/4278/00/4278toc.xml&DOI=10.1109/WSC.1997.640943

Marsden-Smedley, J.B. 1993. Fire characteristics and fire behaviour in Tasmanian buttongrass moorlands. Hobart: Parks and Wildlife Service, Department of Environment and Land Management.

Martell, D. L., K. Hirsch, J. Malcolm, R. McAlpine, A. Wintraub, M. Acuna. 2004. A firesmart approach to integrated fire and forest management in the boreal forest region of canada. Retrieved March 10, 2008, from http://www.sfmnetwork.ca/docs/e/PR_200304-martelldinte6.pdf.

Martell, D. 2007. Forest Fire Management. Handbook Of Operations Research In Natural Resources. Retrieved February 11, 2008, from http://dx.doi.org/10.1007/978-0-387-71815-6_26.

Mladenoff, D.J., G. E. Host, J. Boeder, and T. R. Crow. 1996. LANDIS: a spatial model of forest landscape disturbance, succession, and management. In: Goodchild, M.F., Steyaert, L.T., Parks, B.O. (Eds.), GIS and Environmental Modeling: Progress and Research Issues. GIS World, pp. 175–80.

Morais, M. 2001. Comparing spatially explicit models of fire spread through chaparral fuels: a new model based upon the Rothermel fire spread equation. MA thesis, University of California, Santa Barbara.

Morito, S., K. H. Lee, K. Mizoguchi, and H. Awane. 1993. Exploration of a minimum tardiness dispatching priority for a flexible manufacturing system–A combined simulation/optimization approach, in Proceedings of the 1993 Winter Simulation Conference (Evans, G. W., Mollaghasemi, M., Russell, E. C., and Biles, W. E., eds.), pp. 829–837, Institute of Electrical and Electronics Engineeres, Piscataway, New Jersey.

Natural Resources Canada. 2008. The Atlas of Canada. Retrieved June 5, 2008, from http://atlas.nrcan.gc.ca/site/english/index.html

Nersesian, R. L., M. D. Troutt, and G. J. Weinroth. 2001. Utilizing RISKOptimizer to manage quality performance. Logistics Information Management. 14: 196 - 200.

Noble, I.R., G. A. V. Barry, and A. M. Gill. 1980. McArthur's fire-danger meter expressed as equations. Aust. J. Ecol. 5: 201-203.

Ntaimo, L., B. P. Zeigler, M. J. Vasconcelos, and B. Khargharia. 2004. Forest Fire Spread and Suppression in DEVS. *SIMULATION. 80*: 479-500.

Ólafsson, S., J. Kim. 2002. Simulation optimization. Proc. 2002 Winter Simulation Conf. 79–84.

Ontario Ministry of Natural Resources. 2001. Forest management guide for natural disturbance pattern emulation [online]. Retrieved February 1, 2008, from http://www.web2.mnr.gov.on.ca/mnr/forests/forestdoc/ebr/guide/natural_dist/part%20one.pdf

Omi, P. N. 2005. Forest Fires. Santa Barbara, California: ABC-CLIO

Palma, C.D., W. Cui, D. L. Martell, D. Robak, and A. Weintraub. 2007. Assessing the impact of stand-level harvests on the flammability of forest landscapes. Int. J. of Wildland Fire.16: 584–592.

Parisien, M. A.; Kafka, V. G.; Hirsch, K. G.; Todd, B. M.; Lavoie, S. G.; Maczek, P. D. 2005. Using the Burn-P3 simulation model to map wildfire susceptibility. Information Rep. NOR-X-405. Edmonton, AB: Natural Resources Canada, Canadian Forest Service, Northern Forestry Centre. 36 p.

Parisien, M. A., D. R. Junor, V. G. Kafka. 2006. Using landscape based decision rules to prioritize locations of fuel treatments in the boreal mixedwood of western Canada. USDA Forest Service Proceedings RMRS-P-41.

Payette, S. 1992. Fire as a controlling process in the North American boreal forest. *In* A systems analysis of the boreal forest. Edited by H.H. Shugart, R. Leemans, and G.B. Bonan. Cambridge University Press, Cambridge, U.K. pp. 144–169.

Paulson, E. 1964. A sequential procedure for selecting the population with the largest mean from k normal populations. Ann. of Math. Stat. 35:174-180.

Peet, G. B. 1967. The shape of mild fires in Jarrah forest. Australian Forests. 31: 121-7.

Perera, A.H., D. Yemshanov, F. Schnekenburger, K. Weaver, D. J. B. Baldwin, and D. Boychuk, 2003. Boreal Forest Landscape Dynamics Simulator (BFOLDS): a grid-based spatially stochastic model for predicting crown fire regime and forest cover transition. Forest Research Report No. 152, Ontario Forest Research Institute, Sault Ste. Marie, Ontario, Canada.

Perry, G. L. W. 1998. Current approaches to modelling the spread of wildland fire: a review. Prog. in Phys. Geo. 22: 222–245.

Peter, B. and J. Nelson. 2005. Estimating harvest schedules and profitability under the risk of fire disturbance. Can. J. For. Res. 35:1378–1388.

Pichitlamken, J. and B. L. Nelson. 2003. A combined procedure for optimization via siulation. ACM Transactions on Modeling and Computer Simulation. 13: 155-179.

Pollet, J. and P. N. Omi,. 2002. Effect of thinning and prescribed burning on crown fi re severity in ponderosa pine forests. Int. J. of Wildland Fire. 11: 1-10.

Podur, J., D. L. Martell, and K. Knight. 2002. Statistical quality control analysis of forest fire activity in Canada, Can. J. For. Res. 32: 195– 205.

Prudius, A. A. 2007. PhD Thesis. Retrieved August, 2008, from http://smartech.gatech.edu/handle/1853/16318.

Pyne, S. 1984. Introduction to Wildland Fires: Fire Management in the United States, John Wiley, New York, 1–34.

Reed, W. and D. Errico. (1986). Optimal Harvest Scheduling at the forest level in the presence of the risk of fire. Can. J. For. Res. 16: 266-278.

Richards, G. D. 1990. An Elliptical Growth Model Of Forest Fire Fronts And Its Numerical Solution. International Journal for Numerical Methods in Engineering. 30. 1163-1179.

Rothermel, R. C. 1972. A mathematical model for predicting fire spread in wild land fuels. USDA Forest Service, Research Paper INT-115.

Sargent, R. G. 1991. Simulation model verification and validation. In Proceedings of the 23rd conference on Winter simulation (pp. 37-47). Phoenix, Arizona, United States: IEEE Computer Society.

Schroeder, D. 2006. Effectiveness of forest fuel management: a crown fire case study in the northwest territories, canada (Tech. Rep.). Forest Engineering Research Institute of Canada.

Skinner, W. R., B. J. Stocks, D. L. Martell, B. Bosnal, A. Shabbar. 1999. The association between circulation anomalies in the mid-troposphere and area burned by wildland fire in Canada, Theor. Appl. Climatol. 63: 89– 105.

Skinner, W. R., M. D. Flannigan, B. J. Stocks, D. L. Martell, M. Wotton, J. B. Todd, J. A. Mason, K. A. Logan, and M. Bosch 2002. A 500 hPa synoptic wildland fire climatology for large Canadian forest fires, 1959–1996, Theor. Appl. Climatol. 71: 157– 169.

Sneeuwjagt, R. J. and W. H. Frandsen. 1977. Behavior of experimental grass fires vs. predictions based on Rothermel's fire model. Can. J. For. Res.. 7: 357-367.

Stephens, S. L. 1998. Evaluation of the effects of silvicultural and fuels treatments on potential fire behaviour in Sierra Nevada mixed-conifer forests. For. Ecol. Maange. 105:21–35.

Stocks, B.J. and M. D. Flannigan. 1987. Analysis of the behavior and associated weather for a 1986 northwestern Ontario wildfire: Red Lake No. 7. In Postprint Vol. Ninth Conf. Fire and Forest Meteorology. Am. Meteor. Soc., Boston, Mass. p. 94-100.

Stocks, B.J. 1988. Forest fire close to home: Terrace Bay #7186. In Proc. Symp. and Workshop Protecting People and Homes from Wildfire in the Interior West. USDA For. Serv., Intermt. Res. Stn., Ogden, Utah. Gen. Tech. Rep. INT-251. p. 189-193.

Stocks, B. J., B. D. Lawson, M. E. Alexander, C. E. Van Wagner, R. S. McAlpine, T. J. Lynham, and D. E. Dube. 1989. The Canadian Forest Fire Danger Rating System: An Overview. For. Chron. 65: 450–457.

Stocks, B.J., M. A. Fosberg, T. J. Lynham, L. Mearns, B. M. Wotton, Q. Yang, J. Z. Jin, K. Lawrence, G. R. Hartley, J. A. Mason, and D. W. McKenney. 1998. Climate change and forest fire potential in Russian and Canadian boreal forests. Clim. Change. 38: 1–13.

Swisher, J. R., and S. H. Jacobson. 1999. A survey of ranking, selection, and multiple comparison procedures for discrete-event simulation. In Proceedings of the 31st conference on Winter simulation: Simulation---a bridge to the future - Volume 1 (pp. 492-501). Phoenix, Arizona, United States: ACM.

Thomas, P. H. 1967. Some Aspects of the Growth and Spread of Fire in the Open. Forestry. 40: 139-164.

Tompkins, G. and F. Azadivar. 1995. Genetic Algorithms in Optimizing Simulated Systems, Proceedings of the 1995 Winter Simulation Conference, 757-762.

Tymstra, C. 2002. Prometheus-the canadian wildland fire growth model. Initial Attack , 8-9.

van Wagner, C.E. 1969. A simple fire growth model. Forestry Chron. 45:103-104.

van Wagner, C.E. 1977. Conditions for the start and spread of crownfire. Can. J. For. Res. 7:23-34.

van Wagner, C.E. 1987. Development and structure of the Canadian Forest Fire Weather Index System. Canadian Forest Service, Ottawa, Ont. For. Tech. Rep. 35.

van Wagner, C. E., 1988. The historical pattern of annual burned area in Canada. For. Chron., 64: 182–185; Erratum, 64: 319.

van Wagner, C.E. 1993. Prediction of crown fire behavior in two stands of jack pine. Can. J. For. Res. 23:442-449.

van Wagtendonk, J.W., 1996. Use of a deterministic fire growth model to test fuel treatments. In: Sierra Nevada Ecosystem Project: Final Report to Congress, vol. II. Assessments and Scientific Basis for Management Options. University of California, Davis. Centers for Water and Wildland Resources, pp. 1155–1165

van Wilgen, B.W. and D. M. Richardson. 1985. The effect of alien shrub invasions on vegetation structure and fire behaviour in South African fynbos shrublands: a simulation study. Journal of Applied Ecology 22: 955-966.

Vicars, M. (ed.). 1999. FireSmart: protecting your community from wildfire. Partners in Protection, Edmonton, Alberta.

Vogt, H. 2004. A new method to determine the tool count of a semiconductor factory using FabSim, in Proceedings of the 2004 Winter Simulation Conference (Ingalls, R. G., Rossetti, M. D., Smith, J. S., and Peters, B. A., eds.), pp. 1925–1929, Institute of Electrical and Electronics Engineeres, Piscataway, New Jersey.

Weber, R.O. 1991. Modelling fire spread through fuel beds. Progress in Energy and Combustion Science. 17: 67-82.

Weber, M. G. and B. J. Stocks. 1998. Forest fires and sustainability in the boreal forests of Canada. Ambio: a Journal of the Human Environment. 7: 25-43.

Wei, Y., D. Rideout, and A. Kirsch. 2008. An optimization model for locating fuel treatments across a landscape to reduce expected fire losses. Can. J. For. Res. 38: 868-877.

Wolfram, S. 1984. Computation theory of cellular automata. Comm. in Math. Physic., 96: 15-57.

Wotton, B. M., D. L. Martell, and K. A. Logan. 2003. Climate Change and People-Caused Forest Fire Occurrence in Ontario. Climatic Change, 60: 275-295.

Yang, W., and B. L. Nelson. 1991. Using Common Random Numbers and Control Variates in Multiple-Comparison Procedures. Operations Research. 39: 583-591.

APPENDIX A

FastFire Program Structure

Object	Purpose	Notes
burnCell	Holds all data for each cell in the gridded landscape including: fuel type, age, elevation, value, burned (boolean set to true if affected by fire) and treatment (type of treatment, if any, employed in cell). The burn queue also contains a list of all neighbouring cells.	
burnQueue	A First-In-First-Out linked list of burnCells which holds all cells which are currently burning.	One burn queue is used for each individual fire.

Module	Purpose	Parameters	Returns
void readData(String p)	To read data from text files into memory. Reads in fuel types, age, elevation and value.	String p holds the location of the text data files (e.g. "C:\data")	none
void getIgnitions()	Generates a poisson number of lightning strikes based on a used defined mean and randomly locates them across the landscape. If ignition criteria is met, a new burnQueue is formed and the ignited cell is added to it.	none	none
double getROS(int fuelType)	Returns the front rate of spread for a given fuel type and the current weather conditions as defined in the FBP.	int fuelType holds the FBP code for the given fuel type.	Rate of spread (forward direction)
double getBROS(int fuelType)	Returns the back rate of spread for a given fuel type and the current weather conditions as defined in the FBP.	int fuelType holds the FBP code for the given fuel type.	Rate of spread (backwards direction)
double getSpreadProb (double r)	Converts rate of spread to a probability of cell ignition using a heuristic normalization procedure	double r holds the rate of spread value to be converted	Probability of cell ignition
boolean	Given a cell that is	burnCell start holds	True if the cell

checkEnqueue(burnCell start, burnCell end)	currently burning and a neighbouring cell, this method returns true if the cell will ignite, false otherwise. This is checked by calculating the rate of spread and determining the probability of ignition.	the cell that is currently burning burnCell end hold the neighbouring cell that may ignite given the proper conditions	ignites, false otherwise
void spreadOneDay()	Simulates fire spread for one day by checking the ignition of cells neighbouring currently burning cells iteritively for the user-defined number of time steps in one day.	none	none
void burnOneYear()	For a user-defined number of days in a year this method sets the weather variables from the weather stream, calls getIgnitions and calls spreadOneDay.	none	none
double getDamage()	Calculates the damage done to the landscape by summing the values of each cell that was affected by fire.	none	Total damage done
void setTreatments(int[] d)	Given an array of integers which represent the choice of treatments (1 if the treatment is applied, o otherwise) the correct treatment values are set for the appropriate cells	An array of integers signifying the choice of treatments	none

APPENDIX B

SUMMARY OF CALIBRATION USING HISTORICAL FIRE DATA

Table 6 summarizes the historical data used in the calibration of FastFire. For the years 1973 to 2006, the number of fires and total fire area was determined through the analysis of spatial fire data from an area spanning approximately two million hectares, as shown in Figure 22. All data collected came from a spatial fire perimeter database supplied by the Ontario Ministry of Natural Resources.

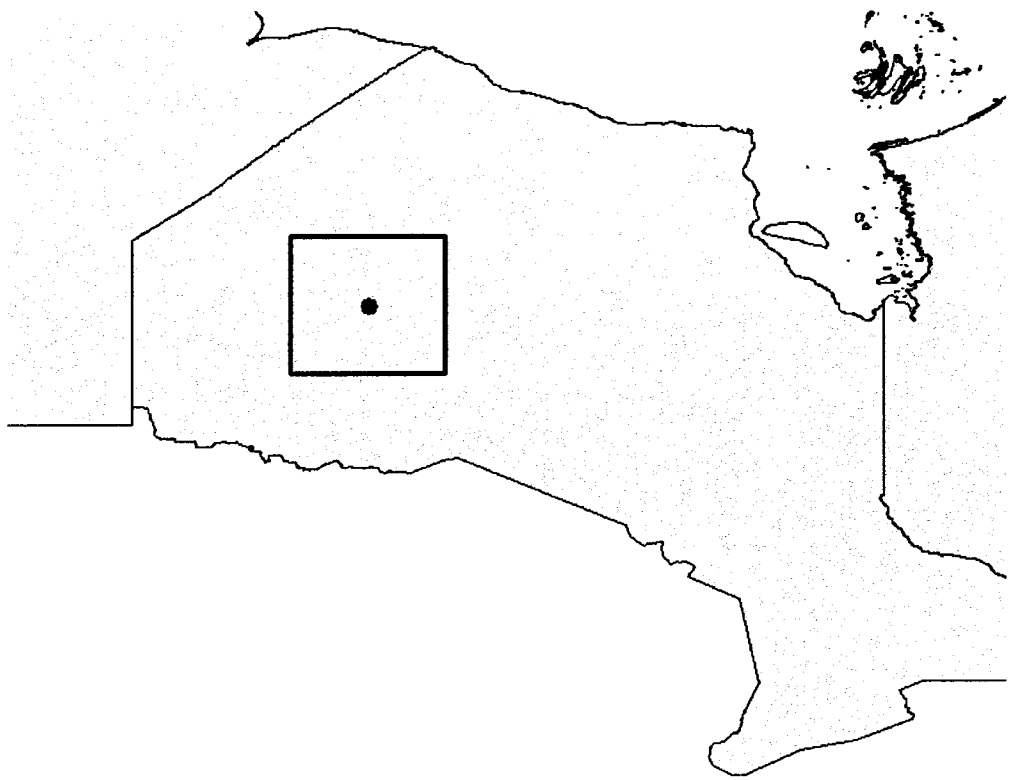

Figure. 22. Area used for calibration of FastFire.

Table 6: Summary of historical fire data spanning 1973-2006 in calibration area.

Year	Number of fires	Total Fire Size
1973	0	0
1974	2	52608
1975	0	0
1976	4	68756
1977	1	72324
1978	0	0
1979	1	1821
1980	3	3493
1981	4	40079
1982	0	0
1983	7	66561
1984	4	10334
1985	0	0
1986	5	5664
1987	1	1445
1988	1	1000
1989	7	31735
1990	9	71189
1991	2	59357
1992	6	100350
1993	1	3750
1994	3	4745
1995	3	17100
1996	6	34340
1997	0	0
1998	2	6524
1999	1	700
2000	0	0
2001	1	2250
2002	1	2600
2003	1	82027
2004	0	0
2005	0	0
2006	0	0
Average	2.235294118	21786.82353