# University of Alberta

A Simulation/Optimization System for Modelling Timber and Old Forest under Stochastic Fire Disturbance.

by

**Matthew David Conrod**

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of

Master of Science
in
Forest Biology and Management

Department of Renewable Resources

**Examining Committee**

Dr. Glen W. Armstrong, Renewable Resources

Dr. Marty K. Luckert, Rural Economy

Dr. Soung-Ryoul Ryu, Renewable Resources

# Abstract

Stochastic wildfire disturbance contributes to uncertainty in forest management planning. In this study, a system composed of an optimizing forest estate model nested within a Monte Carlo simulation model of stand replacing fires is used to investigate the impact stochastic fire may have on the achievement of harvest level and old forest area targets. Two different variations of the modelling system are used to test the impact a buffer stock of timber will have on the probability of achieving these indicators targets. Preliminary results suggest that a reduced harvest level may increase the probability of indicator achievement. However, the immediate harvest level decrease necessary is high and there is still no assurance of target achievement. Further, from a net present value perspective, most scenarios examined showed a higher profit in the absence of a buffer stock.

# Acknowledgements

I would like to thank Dr. Glen Armstrong for his guidance throughout the project and his patience while I attempted to balance my studies with a full time job and a young family. Thank you to Dr. Marty Luckert for taking the time to review the document and participate as a member of my supervisory committee. Thank you to Dave Lindenas and Greg Carlson for allowing me to integrate work and school on occasion. Also, thank you to my family who always provided encouragement and support since the start of my university education. Especially my sister Jacky, who inspired me to enter university after high school. Finally, thank you to my wife and daughters who gave up a lot of irreplaceable family time with me while I pursued this degree.

# Contents

**1 Introduction** **1**

Bibliography . . . 2

**2 Sustainability of timber production and old forest targets with stochastic fire disturbance** **4**

2.1 Introduction . . . 4

2.2 Modeling System . . . 5

2.3 Model Application . . . 10

2.4 Results and Discussion . . . 15

2.5 Conclusions . . . 24

Bibliography . . . 25

**3 An economic evaluation of a standing timber buffer stock strategy** **29**

3.1 Introduction . . . 29

3.2 Modeling System . . . 31

3.3 Model Application . . . 37

3.4 Results . . . 39

3.5 Discussion . . . 39

3.6 Conclusion . . . 45

Bibliography . . . 45

**4 A Case Study Application** **48**

4.1 Introduction . . . 48

4.2 Methodology . . . 48

4.3 Results . . . 57

4.4 Discussion . . . 58

Bibliography . . . 65

**5 Conclusion** **67**

Bibliography . . . 68

**Appendices** **71**

**A Chapter 2 Modeling System in Python Code** **71**

**B Chapter 2 Woodstock Model Formulation** **75**

**C Chapter 3 Modeling System in Python Code** **77**

**D Chapter 3 Woodstock Model Formulation** **84**

**E Chapter 4 Modeling System in Python Code** **88**

**F Chapter 4 Woodstock Model Formulation** **123**

# List of Tables

2.1 Alternative initial age class distributions for deficit, normal, and surplus forests. 12
2.2 Annual area burned and percentage of productive forest burned by year (after adjustment for island remnants). . . . . . . . . . . . . . . . . . . . . . . . . . 14
2.3 Harvest requests for deficit, normal, and surplus forests at different percentages of baseline harvest. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 16

3.1 Alternative initial age class distributions for normal and surplus forests. . . 38
3.2 Distribution of net present values ($10^6$ \$) for discount rates for the normal forest. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 40
3.3 Distribution of net present values ($10^6$ \$) for discount rates for the surplus forest. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 41

4.1 PA FMA area netdown administrative classes . . . . . . . . . . . . . . . . . 51
4.2 PA FMA area netdown stand operability classes . . . . . . . . . . . . . . . 51
4.3 PA FMA area yield groups . . . . . . . . . . . . . . . . . . . . . . . . . . . 52
4.4 Old forest targets . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 55
4.5 Age definitions for old forest . . . . . . . . . . . . . . . . . . . . . . . . . . 55
4.6 Death ages for forest types . . . . . . . . . . . . . . . . . . . . . . . . . . . 56
4.7 Estimated renewal costs for forest types . . . . . . . . . . . . . . . . . . . . 57

# List of Figures


2.1 Overview of Chapter 2 Modelling System . . . . 6
2.2 Location of the Prince Albert Forest Management Area within Saskatchewan, Canada . . . . 11
2.3 Harvest request achievement for normal forest. . . . . 17
2.4 Old forest achievement for normal forest. . . . . 18
2.5 Harvest request achievement for surplus forest. . . . . 20
2.6 Old forest achievement for surplus forest. . . . . 21
2.7 Harvest request achievement for deficit forest. . . . . 22
2.8 Old forest achievement for deficit forest. . . . . 23

3.1 Overview of Chapter 3 Modelling System . . . . 32
3.2 Harvest and old forest results for the no buffer and buffer strategies (Normal forest at a 5% discount rate) . . . . 40
3.3 Harvest and old forest results for the no buffer and buffer strategies (Surplus forest at a 5% discount rate) . . . . 41
3.4 NPV and cumulative 200 year harvest level results for the no buffer and buffer strategies (Normal forest at a 5% discount rate) . . . . 42
3.5 NPV and cumulative 200 year harvest level results for the no buffer and buffer strategies (Surplus forest at a 5% discount rate) . . . . 42

4.1 Overview of Chapter 4 Modelling System . . . . 50
4.2 PA FMA area age class distribution . . . . 53
4.3 Harvest results for the no buffer and buffer strategies . . . . 58
4.4 HW old forest results for the no buffer and buffer strategies . . . . 59
4.5 HJP old forest results for the no buffer and buffer strategies . . . . 59
4.6 HXS old forest results for the no buffer and buffer strategies . . . . 60
4.7 JPH old forest results for the no buffer and buffer strategies . . . . 60
4.8 XSH old forest results for the no buffer and buffer strategies . . . . 61
4.9 BS old forest results for the no buffer and buffer strategies . . . . 61
4.10 JP old forest results for the no buffer and buffer strategies . . . . 62
4.11 JPBS old forest results for the no buffer and buffer strategies . . . . 62
4.12 WSBF old forest results for the no buffer and buffer strategies . . . . 63

4.13 TL old forest results for the no buffer and buffer strategies . . . . . . . . . . 63
4.14 Cumulative harvest and NPV for the no buffer and buffer strategies . . . . 64

# Chapter 1

# Introduction

In the boreal mixedwood forest of Western Canada, forest managers are responsible for making decisions under almost overwhelming uncertainty. Various, largely unpredictable, disturbance processes can have a considerable impact on the forest upon which they are stewards. Among the disturbance processes impacting the boreal mixedwood, wildfire may be considered the dominant stand replacing factor and can be characterized as random (Armstrong, 1999).

Due to this uncertainty, many management plans tend to ignore wildfire altogether. However, in order to foster informed decision making this uncertainty should not be overlooked. Considering this uncertainty in forest management planning can be challenging considering the future forest condition could be one of a large number of possible states. The ability to estimate the probability of a future forest state given a particular management scenario becomes invaluable. In this case the planning process becomes a form of risk assessment. Various management scenarios can be examined by comparing the range of probable forest states that may result.

One of the most crucial decisions a forest manager can make is likely the rate at which timber is harvested. This is the one disturbance factor that they can "control" with some element of certainty. This decision will have direct bearing on the benefits gained from the forest, as well as the future forest state.

This thesis presents three papers that investigate the impacts of stochastic wildfire on multiple values in the boreal mxedwood forest of Saskatchewan, Canada. A probabilistic sustainability approach is taken building on the methodology presented by Armstrong (2004). A simulation/optimization modeling system is used to project the range of probable timber harvest and old forest area levels given a particular management scenario. The simulation/optimization modeling system is composed of a linear programming timber supply model, used to determine optimal harvest plans, nested within a Monte Carlo simulation model of stand replacing fire. The timber harvest level serves as the key policy "lever" among scenarios.

The first paper (Chapter 2) describes a simulation/optimization modelling system similar to that presented in Armstrong (2004). One key difference is the expansion of scope to include a second forest value, area of old forest. A second key difference is in the optimization model formulation. Armstrong (2004) maximized timber production, this system uses a maximin old forest objective function subject to a harvest level constraint. The harvest level constraint was derived from a deterministic baseline model which maximized harvest subject to an old forest constraint in the absence of fire. Succesive runs of the modeling system were completed using reduced proportions of the baseline harvest level as a minimum value constraint. The indicator value distributions (harvest achievement and old forest area) were plotted for each harvest level and compared to assess the effect of a harvest level reduction on indicator achievement.

The second paper (Chapter 3) focuses on using a simulation/optimization modeling system to assess the economic impact of including a buffer stock of standing timber in a management plan. For each planning period, the size of the buffer stock was determined by modeling fire as a deterministic process outside of the modeling system. This approach is sometimes referred to as a model III formulation (Martell, 1994). The resulting model III harvest level was then used as a maximum harvest level constraint within the modeling system. This simulated the cycle of developing a harvest plan with a harvest level that accounts for the mean annual disturbance rate, implementing the harvest plan followed by the"actual" stochastic fire disturbance occuring in the forest. The model objective function was to maximize net present value. The distribution of net present values, harvest achievement and old forest area are plotted and compared for both the presence or absence of a buffer stock.

The third paper (Chapter 4) provides a case study application of Chapter 3 using actual forest inventory data and yield curves taken from the Prince Albert Forest Management Agreement area in Saskatchewan, Canada. Whereas Chapters 3 and 4 provided simple model applications using a hypothetical single species forest, this chapter provides a real world example of a simulation/optimization modeling system.

The concluding chapter summarizes the overall results of the study and provides some suggestions for further research.

# Bibliography

Armstrong, G.W. 1999. A stochastic characterisation of the natural disturbance regime of the boreal mixedwood forest with implications for sustainable forest management. *Can. J. Forest Res.* **29**(4), 424–433.

Armstrong, G.W. 2004. Sustainability of timber supply considering the risk of wildfire. *For. Sci.* **50**(5), 626–639.

Martell, D.L. 1994. The impact of fire on timber supply in Ontario. *For. Chron.* **70**(2), 164–173.

# Chapter 2

# Sustainability of timber production and old forest targets with stochastic fire disturbance

## 2.1 Introduction

The impact of stand-replacing forest fires on sustained yield timber production has been studied by researchers for at least 30 years now. Routledge (1980), Martell (1980), and Reed (1984) have explored the effect of fire risk on the optimal forest rotation. Van Wagner (1983) used a simulation model to evaluate the effect of fire on the sustainable timber harvest level of a forest. Reed and Errico (1986) developed a linear programming based forest management model that incorporates fire using a fixed periodic burn fraction. Gassman (1989) and Boychuk and Martell (1996) developed stochastic programming variants of the forest management model which incorporate distributions of the periodic burn rate into their optimization procedures. Armstrong (2004) developed a system consisting of a deterministic linear programming base forest management model nested within a Monte Carlo simulation of forest fire. He used this to develop estimates of the probability of satisfying a timber sustainability test for each period of a twenty decade simulation. He characterized his approach as "probabilistic sustainability". Peter and Nelson (2005) used a similar approach to evaluate profitability of forest management under risk of fire.

Sustainable forest management (SFM) is currently the guiding paradigm for forest management in Canada and in much of the rest of the world. The Canadian Council of Forest Ministers (CCFM) has developed a criteria and indicators (C&I) framework for SFM based on those developed by the Montréal Process. A major criterion for SFM identified by the Montréal Process and the CCFM is the conservation of biological diversity (biodiversity) (The Montréal Process, 1999; CCFM, 2003). The first element related to biodiversity in both C&I frameworks is ecosystem diversity. The first indicator suggested by the Montréal Process for ecosystem diversity is "area and percent of forest by forest ecosystem type, successional stage, age class, and forest ownership or tenure". In the CCFM framework, the first indicator is "area of forest, by type and age class, and wetlands in each ecozone."

Forest certification schemes intend to ensure that sustainable forest management (SFM) is more than just a platitude. Many forestry operations in Canada have, or are seeking, third-party certification that the forests under their control are being managed sustainably, in order to gain or maintain access to markets and possibly enjoy a price premium for their products. The two major third-party certification standards for sustainable forest management in Canada are those of the Canadian Standards Association (CSA) (CSA, 2002) and the Forest Stewardship Council (FSC Canada, 2004). At the time of this writing more than 74 million ha of forest in Canada are certified by CSA and more than 28 million ha by FSC.

The CSA standard uses the CCFM criteria and its own set of elements in its evaluation of sustainable forest management. Element 1.1 (ecosystem diversity) is "Conserve ecosystem diversity at the landscape level by maintaining the variety of communities and ecosystems that naturally occur in the [defined forest area]." The FSC Boreal Standard uses different criteria (the FSC calls them principles), but indicator 6.3.5 under Principle 6 (environmental impact) requires that

> ... [m]anagement strategies maintain average landscape and/or regional distributions or amounts of the full age-range of old forests identified for a 25% departure from the estimated mean of older forests – in recognition of the range of natural variability, practical constraints and competing objectives. In the absence of a credible estimate of the mean, a minimum of 20% of old forest will be retained.

Area of old forest is an indicator of biodiversity that is straightforward to track in most modern forest management models.

In this paper, we build on Armstrong's (2004) probabilistic sustainability approach in a system that considers both timber production and area of old forest. Like Armstrong, we use a hybrid system comprised of a deterministic forest estate model and a Monte Carlo simulation model. We examine the relationship between harvest levels and the probability of achieving old-forest retention targets at different points over a planning horizon. We use a characterization of the fire regime for a forest management area near Prince Albert, Saskatchewan, Canada and stylized age class distributions to present the model and preliminary conclusions.

## 2.2 Modeling System

### Model overview

The system of models used for this study consists of an optimizing forest estate model used to choose optimal timber harvest plans, nested within a Monte Carlo model of stand-replacing forest fires. The model loops through a number of draws representing possible realizations of the fire regime projected over time. The main outputs of the modelling sytem are projected distributions of harvest levels and area of old forest in each decade of a 200 year projection. The policy lever that we examine is the harvest level request, which is set as a proportion of a baseline harvest schedule.

The initialization stage of the model is used to set a harvest request for the model run. The harvest request is set as a proportion of the baseline harvest level. The model uses the harvest requests and simulated burn projections to develop a projection of the distribution of harvest levels and old-forest area over time. Each draw represents a sequence of decades comprising the projection period. Within a draw, a rolling planning horizon approach is used. A maximin linear program is used to represent the problem. The objective function is set to maximize the minimum area of old-growth forest achieved across all decades of the planning horizon. Constraints are used to ensure that the harvest level across all decades is at the harvest request or greater. Because it is possible that the harvest level constraints result in an infeasible solution, I allow for the downward adjustment of harvest level constraints to ensure that a feasible solution is found. The forest inventory represented in the model is updated to reflect harvest, one period of growth, and stand-replacing fire and the process repeats for the next period. The flowchart in Fig. 2.1 summarizes the process.

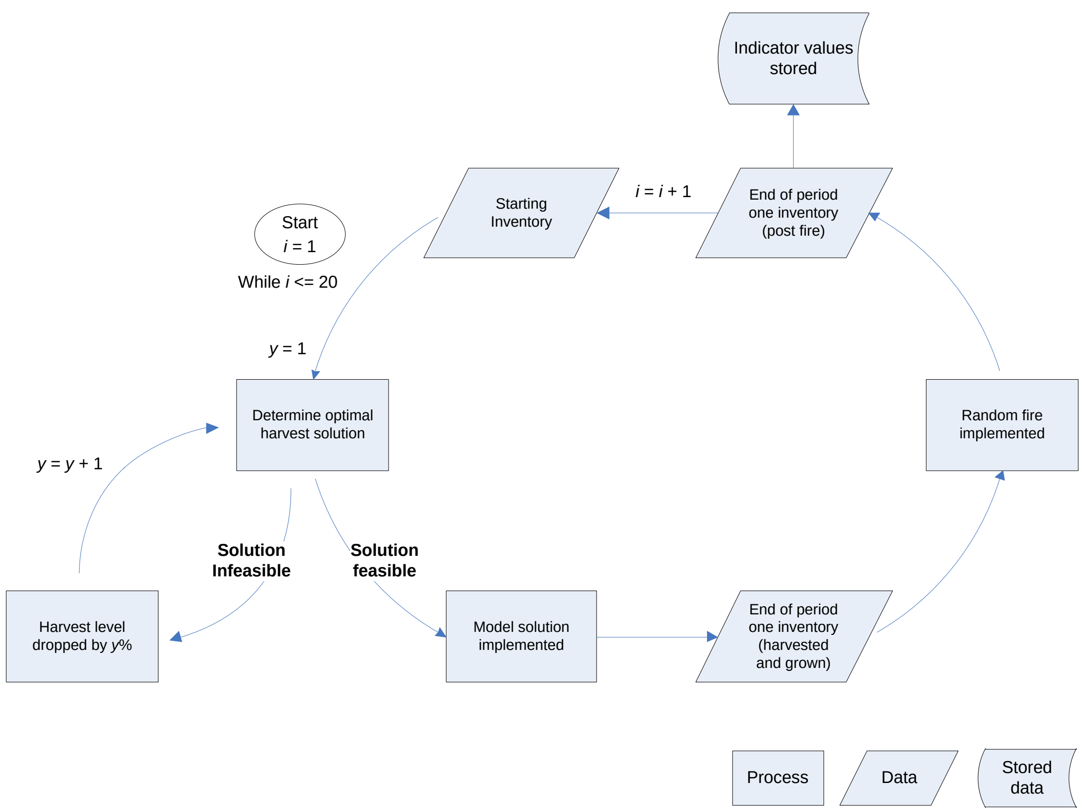



Figure 2.1: Overview of Chapter 2 Modelling System

### Monte Carlo burn proportion model

We assume that the fire regime over the projection period will be the same as that constructed from thirty-one years of historical data. The fire regime is described as a sequence of projected annual burn proportions that are drawn randomly from the fire history. Because the time step in the simulation loop is one decade, these annual projections of burn proportions need to be converted into decadal proportions.

The burn rate for each decade is calculated as

$$\Lambda = 1 - \prod_{i=n}^{n} (1 - \lambda_i) \tag{2.1}$$

where $\lambda_i$ is the randomly drawn annual burn rate. The decadal burn rate, $\Lambda$, is the proportion of the forest area burned at least once in the decade.

### Deterministic baseline

The baseline timber harvest level is defined to be the maximum timber harvest that could be supported by the forest over a planning horizon comprising a specified number of periods, subject to:

1. equal harvest volumes in each period,
2. non-declining growing stock volume in a specified number of periods at the end of the planning horizon, and
3. the maintenance of old-forest area above a threshold in specified periods of the planning horizon.

This is a reasonably accurate simplification of the usual forest management problem on public forest land in most of Canada, with the addition of old-forest area constraints. The stands in the forest are assumed to be managed using a clearcut harvest and plant, even-aged silvicultural system. Harvest volume is produced as a result of timber harvest, and the age of the harvested area is set to zero immediately following harvest. The stands in the forest grow according to a yield table, which specifies the standing timber volume per unit area for stands as a function of age. The yield function also specifies the harvest volume at the age of harvest. No unplanned disturbance such as stand-replacing fire was taken into account.

The linear programming model was built using the Woodstock forest modeling software (Remsoft, 2009) and solved using MOSEK Optimization Tools (MOSEK, 2009). Woodstock represents the problem using a variant of the model II harvest scheduling formulation described by Johnson and Scheurman (1977). The problem is presented below in mathematial notation following Dykstra (1984).

The objective function maximizes the total volume harvested over a planning horizon comprising $H$ periods. In the model II formulation, it is necessary to track "birth" and "harvest" periods for stands in the forest.

$$\max Z = \sum_{j=1}^{H} \sum_{i=-M}^{j-1} v_{ij} x_{ij} \tag{2.2}$$

The birth period for the oldest existing stand in the forest is period $-M$. The birth period for a one-period old stand at the beginning of the planning horizon is period 0. The decision variables $x_{ij}$ represent the area of forest born in period $i$ harvested in period $j$. The yield table for the forest is represented by $v_{ij}$ which represents the volume per ha harvested from a stand born in period $i$ and harvested in period $j$. The age at harvest (in periods) can be calculated as $j - i$.

Area constraints are needed in the model II formulation to ensure that all of the forest existing at the beginning of the planning horizon is assigned to a harvest or a no harvest activity, and to ensure that area born in each period of the planning horizon equates to the area harvested. In Eq. 2.3, $A_i$ represents the area of forest belonging to each birth period at the start of the problem. The decision variables, $w_i$, represent the area from each of these birth periods left unharvested at the end of the planning horizon. Eq. 2.3 ensures that all of the area in each of the inital birth periods is assigned to a valid harvest or no harvest decision. Eq. 2.4 is used to ensure that the area of forest harvested in each period, $j$, of the planning horzion is equal to the area of forest harvested. In this way, the forest is assumed to regenerate immediately following harvest.

$$\sum_{j=1}^{H} x_{ij} + w_i = A_i \qquad i = -M, -M+1, \ldots, 0 \tag{2.3}$$

$$\sum_{k=j}^{H} x_{jk} + w_j - \sum_{i=-M}^{j-1} x_{ij} = 0 \qquad j = 1, 2, \ldots, H \tag{2.4}$$

Eq. 2.5 is used to calculate the timber harvest volume, $F_k$ in each period of the planning horizon. Eq. 2.6 is the even flow constraint which ensures that the harvest volume in each period equals that of the previous period.

$$F_k - \sum_{i=-M}^{k} v_{ik} x_{ik} = 0 \qquad k = 1, 2, \ldots, H \tag{2.5}$$

$$F_k - F_{k-1} = 0 \qquad k = 2, 3, \ldots, H \tag{2.6}$$

The next block of constraints is used to calculate the age class distribution in each period, after harvest, but before growth. $D_{jk}$ represents the area of forest in age class $j$ in period $k$. Eq. 2.7 summarizes the area harvested in each period or, equivalently, the area of forest zero periods old, in each period of the planning horizon. Eq. 2.8 calculates the area in all the other age classes.

$$D_{0k} - \sum_{j=-M}^{k} x_{ik} = 0 \qquad k = 1, 2, \ldots, H \tag{2.7}$$

$$D_{k-i,k} - \sum_{i=-M}^{k} \sum_{j=k+1}^{H} x_{ij} + w_i = 0 \qquad k = 1, 2, \ldots, H \tag{2.8}$$

The variable $E_k$ is used to represent the area of old growth in period $k$. In Eq. 2.9, the constant $U$ represents the youngest age class considered to be old growth. Eq. 2.10 is the constraint used to ensure that old growth area in periods exceeds the desired threshold, $\bar{E}$. In cases where the initial condition of the forest has less old growth area than the threshold, we can set the beginning period for the set of constraints, $s$, to be later than the first period to allow for a solution.

$$E_k - \sum_{e=U}^{H+M} D_{ek} = 0 \qquad k = 1, 2, \ldots, H \tag{2.9}$$

$$E_k \geq \bar{E} \qquad k = s, s+1, \ldots, H \tag{2.10}$$

In order to ensure that the model does not liquidate the forest at the end of the planning horizon, we implement constraints that ensure that the volume of growing stock is non-declining over a period at the end of the planning horizon. Eq. 2.11 summarizes the growing stock volume, $G_k$, in each period $k$. Eq. 2.12 is non-declining from period $Q$ to the end of the planning horizon, $H$.

$$G_k - \sum_{j=k}^{H} x_{ij} v_{ik} + w_i v_{ik} = 0 \qquad k = 1, 2, \ldots, H \tag{2.11}$$

$$G_k - G_{k-1} \geq 0 \qquad k = Q, \ldots, H \tag{2.12}$$

The non-negativity constraints represented in Eq. 2.13 ensure that all of the decision variables in the program take on non-negative values.

$$x_{ij}, w_i, F_k, E_k, G_k, D_k \geq 0 \qquad \forall i, j, k \tag{2.13}$$

### Maximin formulation

Within the simulation cycle, the linear programming model takes on the form of a maximin programming problem (Kawaguchi and Maruyama, 1976). The objective function maximizes the minimum area of old growth over the periods of the planning horizon (Eq. 2.14). Fundamental to this maximin problem is a constraint, as specified in Eq. 2.15, which ensures that $E_{\min}$ represents the area of old-forest in the period with the smallest old-forest area. The parameter $h$ represents the first period in which the old-forest area constraints are applicable.

$$\max E_{\min} \tag{2.14}$$

$$E_{\min} - E_k \leq 0 \qquad k = h, h+1, h+2, \ldots, H \tag{2.15}$$

If one were to transform this constraint by bringing $E_k$ to the right hand side, the mechanics of the maximin problem may become clear (Eq. 2.16). $E_{\min}$, which is being maximized in our objective function, must be less than or equal to the old forest area in all periods upon which our constraint is applied (in our case all periods in the planning horizon). With this constraint, the period(s) with the lowest old forest area will become the value being maximized.

$$E_{\min} \leq E_k \qquad k = h, h+1, h+2, \ldots, H \tag{2.16}$$

The minimum old-forest area constraints specified in Eq. 2.10 are not relevant to the maximin problem and are not used here. A constraint represented by Eq. 2.17 is added to ensure that volume harvested in each period meets the harvest volume request, $\bar{F}$.

$$F_k - \bar{F} = 0 \qquad k = 1, 2, \ldots, H \tag{2.17}$$

## 2.3 Model Application

To provide a demonstration of the methodology described in this paper a stylized model was developed based on the productive forest area and fire regime from the Prince Albert Forest Management Agreement (FMA) area in Saskatchewan, Canada. The Prince Albert FMA occupies 3,290,827 ha roughly in the geographic centre of Saskatchewan (Fig. 2.2). About 97% of the FMA area falls within the Boreal Plains ecoregion (Acton *et al.*, 1998). In general, the boreal plains have good commercial forestry potential relative to other ecozones in Saskatchewan. The forest is predominantly composed of closed-crown mixedwood and conifer stands. The dominant softwood species include white spruce (*Picea glauca* (Moench) Voss), black spruce (*Picea mariana* (Mill.) BSP), jack pine (*Pinus banksiana* Lamb.) and tamarack (*Larix laricina* (Du Roi) K.Koch) while typical deciduous species include trembling aspen (*Populus tremuloides* Michx.), white birch (*Betula papyrifera* Marsh.) and balsam poplar (*Populus balsamifera* L.). The remainder of the FMA (3.3%) falls within the Boreal Shield ecoregion.

Portions of the study area have been subject to timber extraction as early as the 1880s with the establishment of a number of local lumber companies. By 1900, the region had become a relatively significant producer of lumber in western Canada (Walker *et al.*, 1996). Forestry continued to be a key economic component of the region throughout the twentieth century. When all facilities were operating, the study area provided secure wood flow

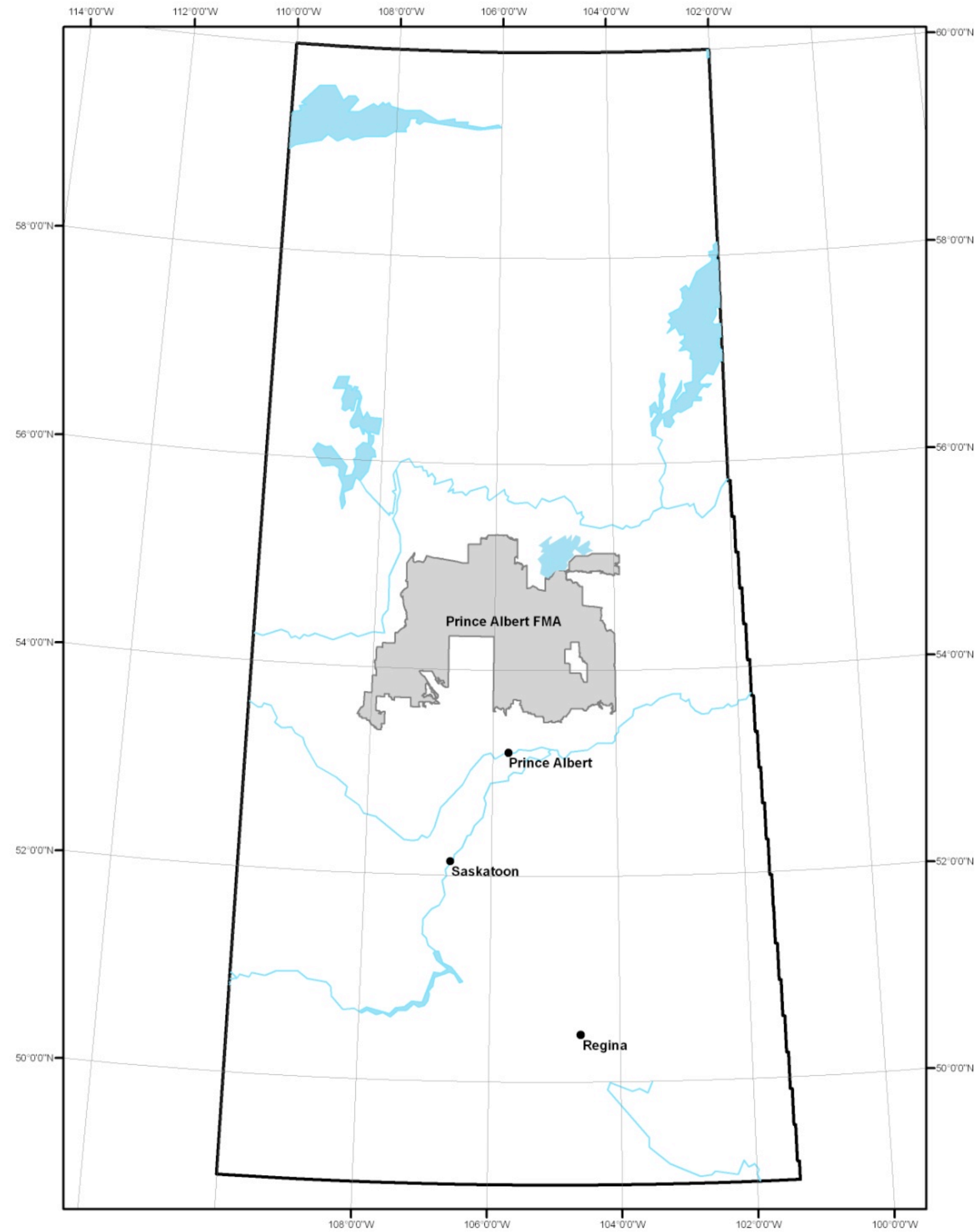



Figure 2.2: Location of the Prince Albert Forest Management Area within Saskatchewan, Canada

to two sawmills, a pulp mill and a variety of other small scale operations such as fence post treatment facilities. Due to a prolonged economic downturn in the forest industry, the majority of these facilities have shut down in recent years. Timber harvest has been significantly below the sustainable potential for the region.

### Input Data

The input data for this analysis are similar to those used by Armstrong (2004). Three hypothetical single species forests with deficit, normal and surplus starting inventory age class distributions were used. The total operable area in each hypothetical forest matched the total operable area for the Prince Albert FMA area and was split equally into age classes according to Table 2.1. The three hypothetical forests provide an approximation of the range of age classes that may occur on forested landscapes in the boreal forest.

Table 2.1: Alternative initial age class distributions for deficit, normal, and surplus forests.

| Age Class (yr) | Deficit (% area) | Normal (% area) | Surplus (% area) |
|---:|---:|---:|---:|
| 10 | 20 | 10 | 0 |
| 20 | 20 | 10 | 0 |
| 30 | 20 | 10 | 0 |
| 40 | 20 | 10 | 0 |
| 50 | 20 | 10 | 0 |
| 60 | 0 | 10 | 10 |
| 70 | 0 | 10 | 10 |
| 80 | 0 | 10 | 10 |
| 90 | 0 | 10 | 10 |
| 100 | 0 | 10 | 10 |
| 110 | 0 | 0 | 10 |
| 120 | 0 | 0 | 10 |
| 130 | 0 | 0 | 10 |
| 140 | 0 | 0 | 10 |
| 150 | 0 | 0 | 10 |

As in Armstrong (2004), stands were assumed to grow according to a Chapman-Richards yield function, $y(t) = b_1 \left(1 - e^{b_2 t}\right)^{b_3}$, where $t$ is age in years, $y$ is the timber yield ($\mathrm{m}^3\ \mathrm{ha}^{-1}$), and $b_1$, $b_2$ and $b_3$ are coefficients with the values 200, -0.026604, and 5 respectively. These coefficients were chosen such that the maximum yield is 200 $\mathrm{m}^3\ \mathrm{ha}^{-1}$ and the mean annual increment (MAI) reaches a maximum at 100 years of age. The maximum MAI for this yield curve is 1.392 $\mathrm{m}^3\ \mathrm{ha}^{-1}\ \mathrm{yr}^{-1}$. It is assumed that both harvest and fire immediately reset the age of the stand to zero, and that the regenerated stands follow the same yield curve.

### Fire Regime Model

For the stochastic component of this hybrid system, I constructed a model of the forest fire regime for the study area. Following Armstrong (1999), I assume that the fire regime for the study area can be described as a random draw from a distribution of proportions of the study area burned annually. Unlike Armstrong, I do not try to estimate the parameters of a lognormal or any other continuous distribution. The simulated annual burn rates are randomly drawn from a list of historical annual burn rates observed on the study area.

There are two major reasons I did not use a lognormal distribution for this study: the distribution does not allow for years with zero area burned and it has no upper bound. The Prince Albert FMA fire history data for the 31-year period used has 9 years where no fires were recorded (Table 2.2). This clearly presents a problem for the estimation of parameters for a lognormal distribution. Armstrong (2004) handled the upper bound issue by truncating the simulated annual burn proportion at 0.20. I decided to draw the annual burn rates directly from the observed distribution. Because of the short time span of the fire history used, it is likely that the full range of annual area burned in the study area is not represented, especially at the upper end. However, this approach is better, in my opinion, than applying arbitrary adjustments to the data and arbitrary truncation points to the observed distribution.

The annual-area burned summary in Table 2.2 was derived from an updated version of the Forest Fire Chronology of Saskatchewan (FFCS) dataset. The Wildlife Branch of Saskatchewan Environment and Resource Management (SERM) developed the initial version as a spatial polygon database of forest fires bigger than 100 ha dating from 1945-1996 (Naelapea and Nickeson, 1998). This dataset has been updated annually by SERM's Forest Fire Protection Branch and was current up to the 2007 fire season during model development. The fire data represents all fires regardless of cause and the burn proportion is calculated as the FMA area burned from fires that started both within and without the FMA area.

The FFCS fire data spans sixty-seven years. I restricted our characterization to the last thirty-one years from 1977 to 2007. Rock (1996) expresses concern that some large fires may be missing from the FFCS for the period 1972-1976 despite efforts undertaken to correct the database. It is also suspected that some large fires are missing from the database for years prior to 1970. In addition to these concerns, the early 1970s saw the adoption of many of the fire management techniques currently used in Saskatchewan (Rock, 1996). Because of changes in fire management techniques, the pre-1970 fire regime could be quite different from the current one.

Various attempts have been made to account for the effect of fire protection activities on annual area burned (Murphy, 1985; Cumming, 1997, 2005). Further, although it seems logical that fire protection practices would reduce the amount of area burned, the real effectiveness of initial attack and fire suppression policies are still debated (Miyanishi and

Table 2.2: Annual area burned and percentage of productive forest burned by year (after adjustment for island remnants).

| Year | Burned Area (ha) | Productive Forest Burned (% area) |
|---|---|---|
| 1977 | 44,739 | 2.59 |
| 1978 | 0 | 0 |
| 1979 | 0 | 0 |
| 1980 | 24,904 | 1.44 |
| 1981 | 876 | 0.05 |
| 1982 | 0 | 0 |
| 1983 | 0 | 0 |
| 1984 | 3,736 | 0.22 |
| 1985 | 0 | 0 |
| 1986 | 0 | 0 |
| 1987 | 54,262 | 3.14 |
| 1988 | 19,658 | 1.14 |
| 1989 | 13,136 | 0.76 |
| 1990 | 1,301 | 0.08 |
| 1991 | 536 | 0.03 |
| 1992 | 0 | 0 |
| 1993 | 2,871 | 0.17 |
| 1994 | 294 | 0.02 |
| 1995 | 111,777 | 6.47 |
| 1996 | 1,702 | 0.1 |
| 1997 | 0 | 0 |
| 1998 | 3,283 | 0.19 |
| 1999 | 2,267 | 0.13 |
| 2000 | 0 | 0 |
| 2001 | 531 | 0.03 |
| 2002 | 24,001 | 1.39 |
| 2003 | 10,292 | 0.6 |
| 2004 | 0 | 0 |
| 2005 | 5,585 | 0.32 |
| 2006 | 5,983 | 0.35 |
| 2007 | 0 | 0 |
| Mean | 10,718 | 0.62 |

Johnson, 2001; Bridge *et al.*, 2005; Cumming, 2005).

The underlying assumption of the characterization of the fire regime used here is that the distribution of annual area burned in the period 1977-2007 represents the fire regime for the period of projection. Although climate change studies have suggested an increase in fire frequency is possible in the future (Johnson *et al.*, 1999), the impact of this increase on old forest retention targets will not be addressed in this paper.

Various methods of data collection were used during the period covered by the FFCS dataset including simple sketches in the early years to GPS and satellite imagery in recent years. With the exception of 2007, only the fire perimeter was mapped for each burn patch. Wetlands, waterbodies and other non-productive polygons within the fire perimeter were not identified. To account for these areas within each burn patch only forested or land capable of supporting forest was used to calculate total area burned. The Saskatchewan Forest Vegetation Inventory (photo source year 1997-2003) for the Prince Albert FMA was used to identify productive polygons.

Typically, fires do not burn all of the productive forest within a fire perimeter (Kachmar and Sanchez-Azofeifa, 2006). Therefore a simple polygon representing the outer perimeter of a fire would overestimate area burned. Fire behavior is complex and is influenced by numerous factors including topography, climate and vegetation type (Turner and Romme, 1994). The resulting burn pattern will tend to be irregular with live islands of forest interspersed within the disturbance patch (Eberhart and Woodard, 1987).

In a detailed analysis of fire patterns in Saskatchewan, Andison (2006) found that the amount of surviving island remnant after a fire varied widely but averaged 24% of the total event area. Of this 24% of the fire event area only 7% of island remnants were fully intact (100% tree survival) and 33% had high-survival (75-99% tree survival). The remainder of the island remnants had less than 75% percent of the trees survive the fire event. Based on these numbers a total of 40% of the island remnants had "full" or "high" survival which equates to 9.6% of fire area calculated using the fire perimeters. For this study the annual area burned calculated based on fire perimeters was reduced by 9.6% to reflect survival within the fire perimeter.

Table 2.2 presents the adjusted fire history data for the Prince Albert FMA. The mean annual area burned in the period is 0.62% of the total productive forest area. This figure is consistent with mean annual burn rate of 0.59% estimated by Parisien *et al.* (2004) for the mid-boreal upland ecoregion. The annual burn percentage is variable, ranging from zero to nearly 6.5 percent. The years 1977, 1987, and 1995 could be considered episodic fire years with large areas burnt.

## 2.4 Results and Discussion

The results from the baseline models show that for the deficit, normal, and surplus forests decadal harvest volumes of 11.6, 24.1, and 24.5 million $m^3$ could be considered sustainable (over 200 years) if fire disturbance is not taken into account. Table 2.3 presents the initial harvest requests for 100%, 75%, 50%, and 25% of baseline harvest levels examined in this paper. The difference between the baseline and harvest requests represent an attempt to provide a timber buffer to account for the impact of fire.

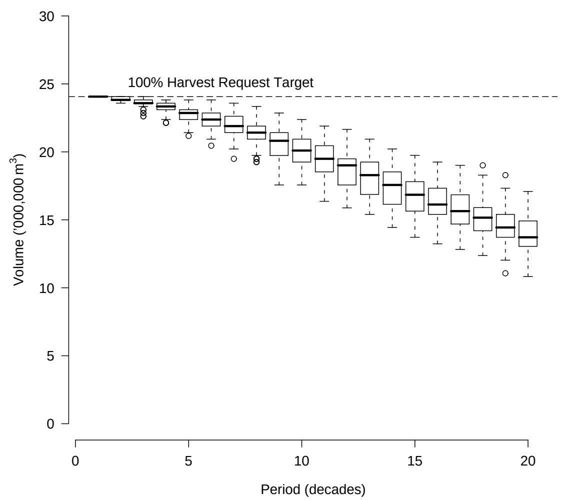
Table 2.3: Harvest requests for deficit, normal, and surplus forests at different percentages of baseline harvest.

| Proportion | Deficit Forest ($10^6$ $m^3$/decade) | Normal Forest ($10^6$ $m^3$/decade) | Surplus Forest ($10^6$ $m^3$/decade) |
|---|---|---|---|
| 100% | 11.6 | 24.1 | 24.5 |
| 75% | 8.7 | 18.0 | 18.4 |
| 50% | 5.8 | 12.0 | 12.3 |
| 25% | 2.9 | 6.0 | 6.1 |

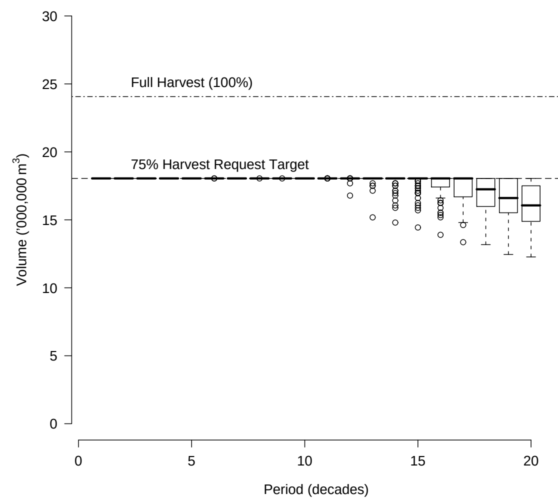
Figs. 2.3 and 2.4 summarize the results of the modelling system for the normal forest. Fig. 2.3 shows the distribution of achieved harvest levels for each of the four harvest level requests. The distribution of achieved harvest levels in each period is represented with a boxplot. The median, upper and lower quartiles are represented by the box. The whiskers represent the largest or smallest non-outlier value and the circles represent outliers. Unsurprisingly, Fig. 2.3a demonstrates that the 100% harvest request is clearly unsustainable very early in the planning horizon when fire occurs on the landscape. Harvest levels less than the initial harvest request are a result of the harvest request reduction procedure used to avoid infeasiblity. The 75% harvest request appears to be "mostly sustainable" until about year 160 in the simulation period when the median harvest level dips noticeably below the harvest request (Fig. 2.3b). The 50% (Fig. 2.3c) and 25% (Fig. 2.3d) harvest requests appear to be sustainable over the simulation period.

With respect to achievement of the old forest area goals on the normal forest, the 100% harvest request level leads to complete failure. After the first period, the entire distribution of outcomes fall below the threshold, and gets steadily worse over the simulation horizon (Fig. 2.4a). With the 75% harvest request (Fig. 2.4b), the distribution of old-forest area above the threshold is maintained for about 7 decades. In the seventh decade, about 25% of the simulations have areas of old-growth below the threshold, and by the ninth period more than 50% of the simulations are below the threshold. Old-forest area is sustainable for longer under the 50% harvest request. Before the 15th decade, more than 50% of the simulations exceed the old-forest threshold (Fig. 2.4c). The lowest harvest request, 25%, has better results as far as the old-forest targets go. By the end of the simulation period, only about one-quarter of the simulations did not meet the target.

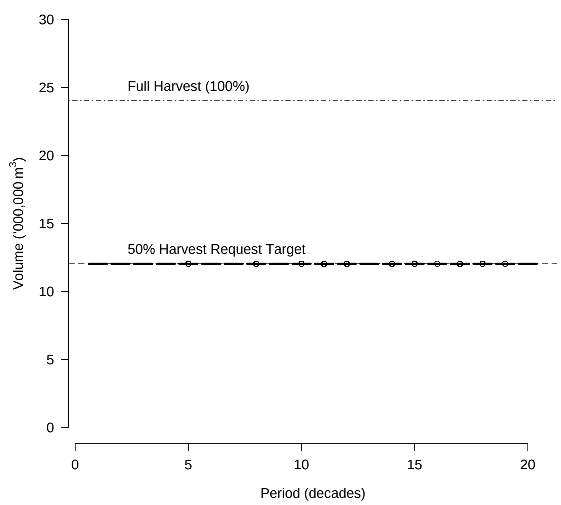
Figs. 2.3 and 2.4 illustrate the difficulties facing a decision maker when stochastic events such as stand-replacing fire are taken into account. The pass-fail line is not clear when results are presented as probability distributions. Is a 50% probability of meeting a sustainability criterion 80 years in the future acceptable? What is the appropriate way of trading off the



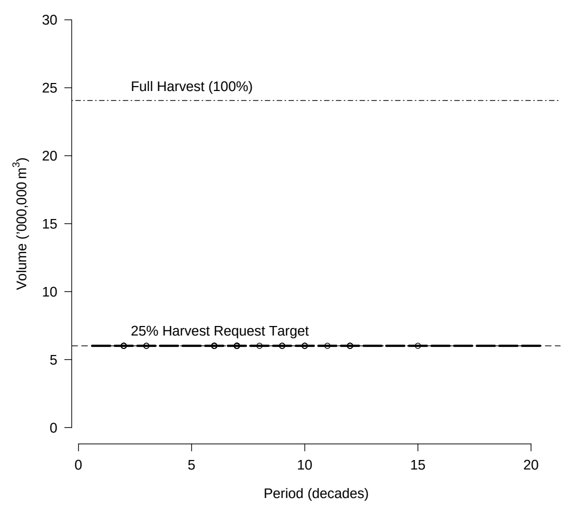

(a) 100% of Baseline Harvest Request




(b) 75% of Baseline Harvest Request

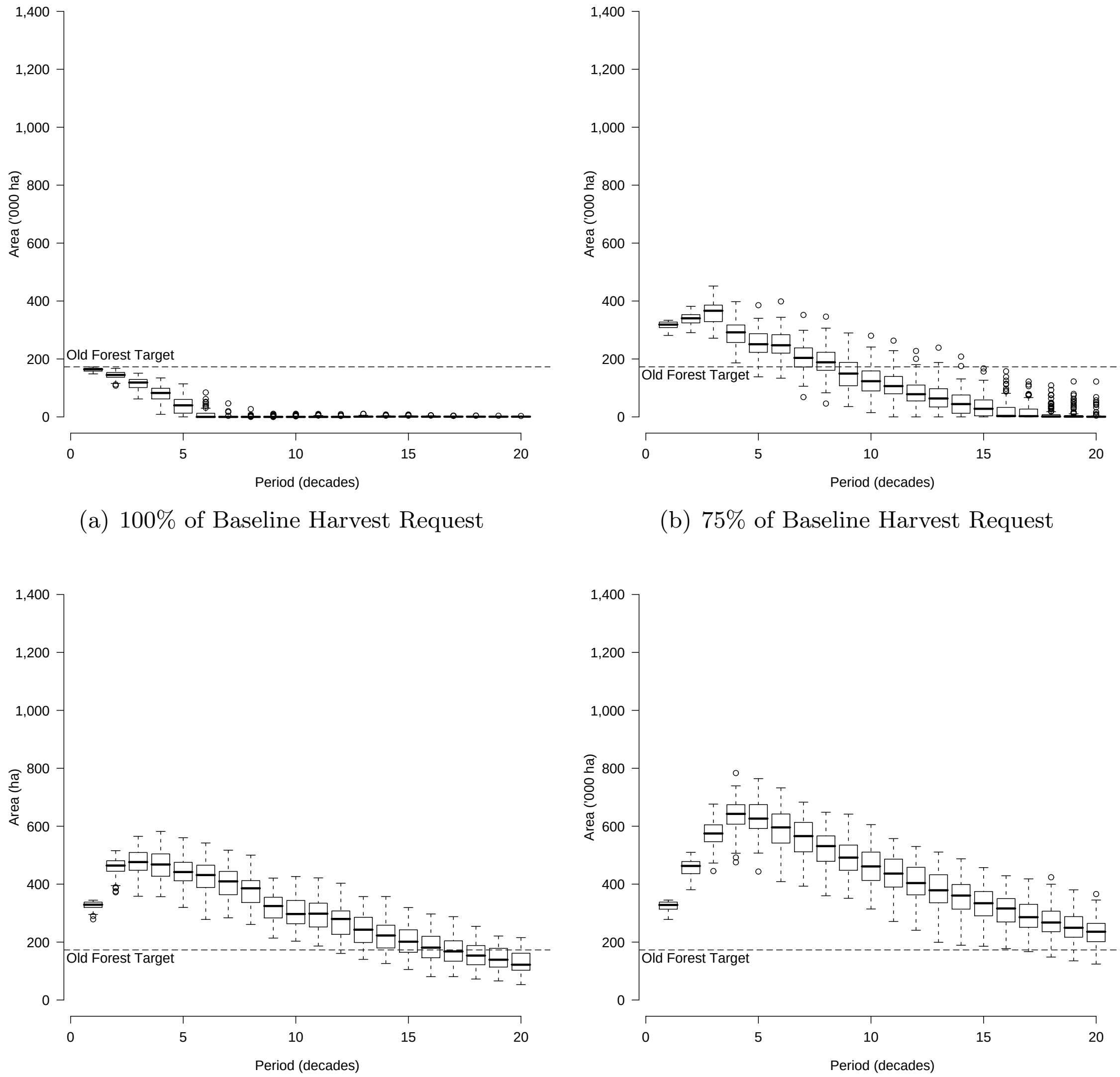



(c) 50% of Baseline Harvest Request




(d) 25% of Baseline Harvest Request

Figure 2.3: Harvest request achievement for normal forest.




(a) 100% of Baseline Harvest Request

(b) 75% of Baseline Harvest Request

(c) 50% of Baseline Harvest Request

(d) 25% of Baseline Harvest Request

Figure 2.4: Old forest achievement for normal forest.

improvements in probability distributions against the certainty of an immediate decrease in harvest and the associated economic benefits? These choices are difficult to make, but for the sake of argument in this paper we will deem a solution that has more than a 50% probability of either a harvest level or old-forest area less than the threshold at any point in the projection horizon to be unsustainble. For the normal forest, the 75% harvest request is unsustainable with respect to both the harvest level and old-forest area thresholds. The 50% request is unsustainable with respect to old-forest area. The 25% harvest request is sustainable using both indicators. In order to meet both harvest and old-forest area criteria under a stochastic fire regime, the harvest request must be set to less than 50% of what is indicated by the deterministic solution.

The results for the surplus forest (Figs. 2.5 and 2.6) show the same general results, but because the forest is older to begin with, old-forest targets are easier to achieve. A 75% of baseline harvest request leads to unacceptable old forest outcomes in the 18th decade and later. This means that the harvest level should be reduced by more than 25% from the deterministic baseline in order to meet harvest level and old-forest area indicators.

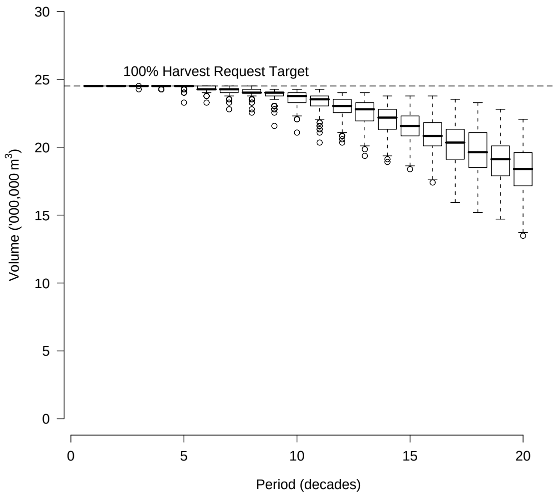
The results for the deficit forest (Figs. 2.7 and 2.8) are somewhat different. Because of our definition of old-forest and the starting age class structure for the deficit forest, there is no old-forest area present at the start of the modelling horizon. This is evident in the first four periods of the graphs in Figure 2.8. The graphs do not indicate the presence of old-forest until the fifth decade.

With respect to the old-forest distributions beginning in the fifth decade, harvest requests of 100% and 75% are clearly unsustainable. The 50% harvest request has median old-forest areas slightly below the threshold starting in period 18. With the 25% harvest request, the median old-forest areas were well above the thresholds for the entire simulation period.

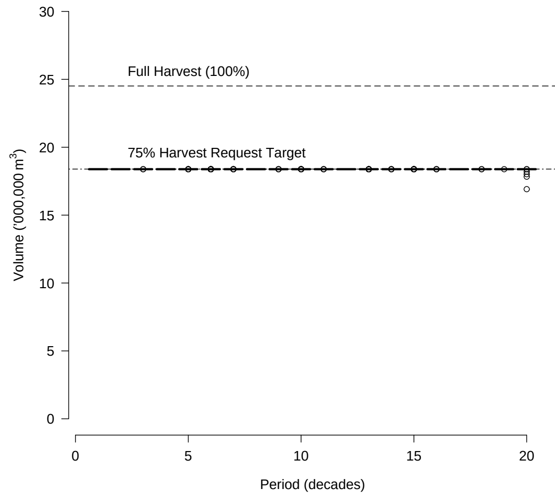
All of the harvest request levels except the last period of the 100% are sustainable on the basis of the harvest level indicator. Perhaps a non-declining yield baseline would be more appropriate for the deficit forest.

The introduction of stochastic fire disturbance has a considerable constraining effect on the achievement of both the timber supply and old-forest targets as expected based on work from other analyses (Klenner *et al.*, 2000; Fall *et al.*, 2004). This exercise suggests that in a stochastic system there is little assurance of meeting the harvest volume or old-forest targets.

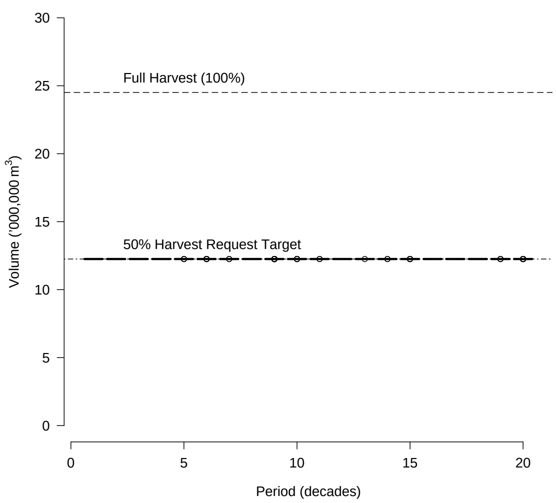
Similar to what others have indicated (Boyland *et al.*, 2005), reducing the harvest level does appear to increase the probability of achieving and maintaining the desired forest values examined. Due mainly to the modeling formulation used, the harvest reduction is particularly effective at improving the probability of maintaining a harvest level request. This could be attributed to the fact that the harvest level request was formulated as an LP constraint in the maxmin formulation. Of course, the model would need to satisfy this



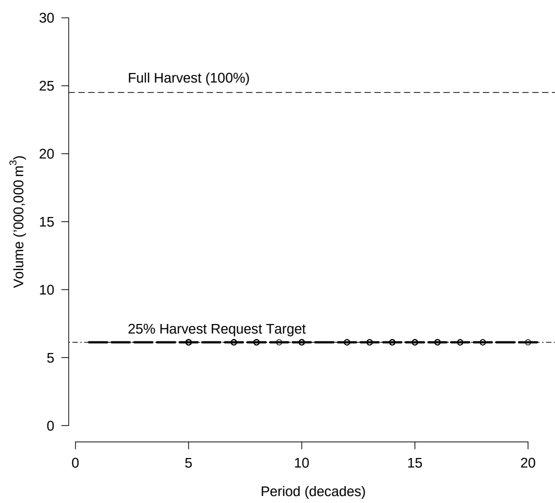

(a) 100% of Baseline Harvest Request




(b) 75% of Baseline Harvest Request

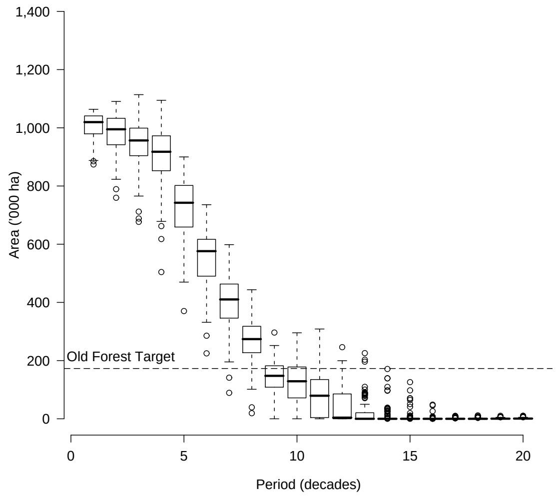



(c) 50% of Baseline Harvest Request

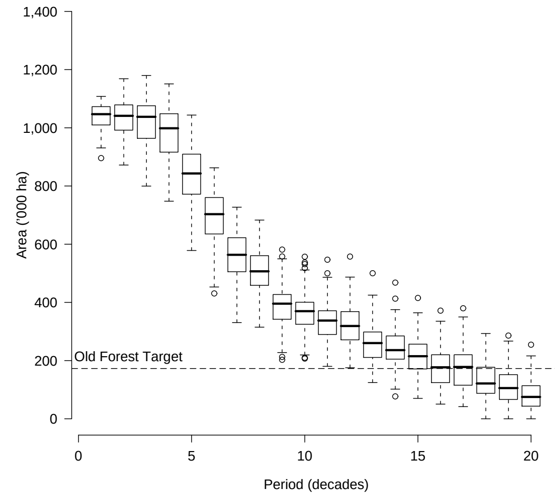



(d) 25% of Baseline Harvest Request

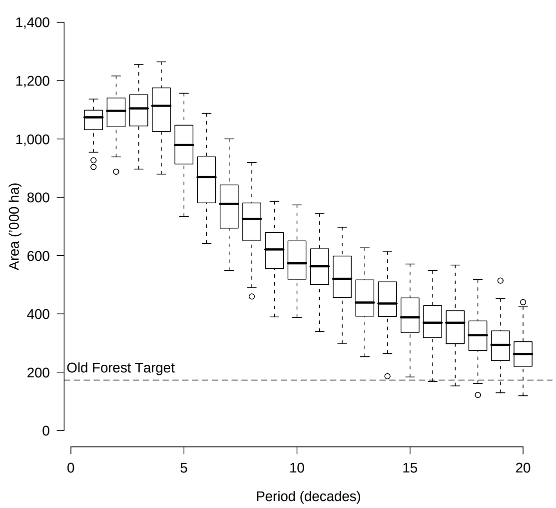
Figure 2.5: Harvest request achievement for surplus forest.



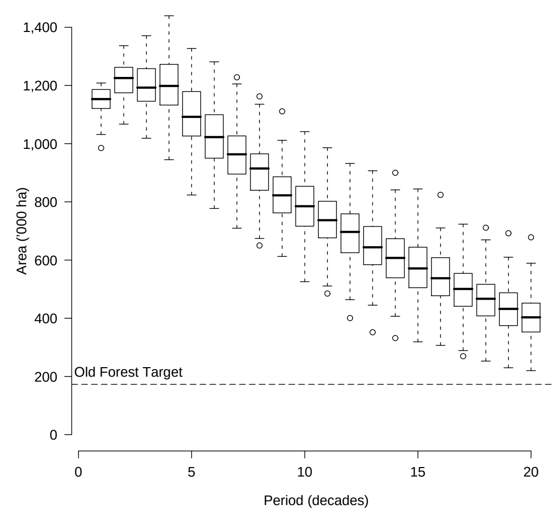

(a) 100% of Baseline Harvest Request




(b) 75% of Baseline Harvest Request

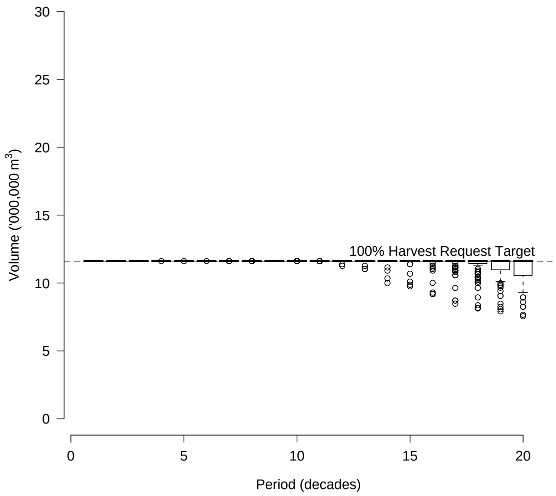



(c) 50% of Baseline Harvest Request

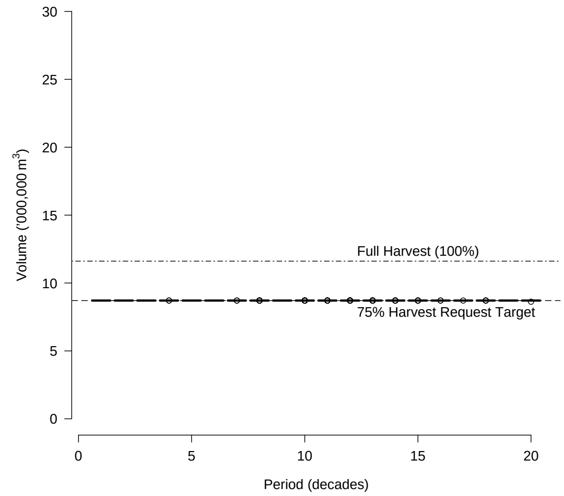



(d) 25% of Baseline Harvest Request

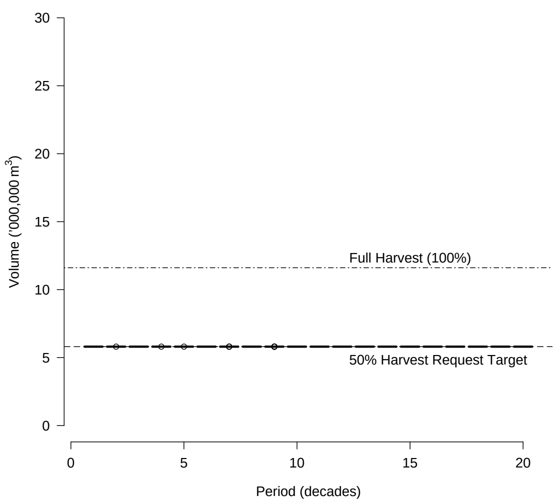
Figure 2.6: Old forest achievement for surplus forest.



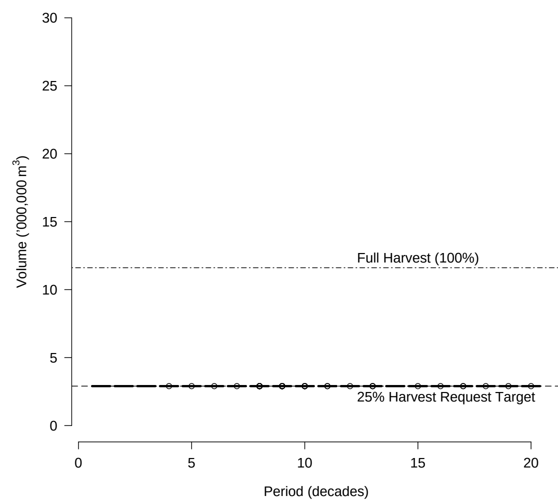

(a) 100% of Baseline Harvest Request




(b) 75% of Baseline Harvest Request

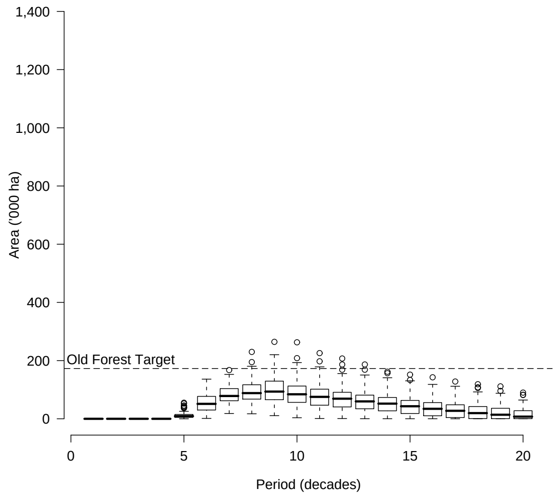



(c) 50% of Baseline Harvest Request

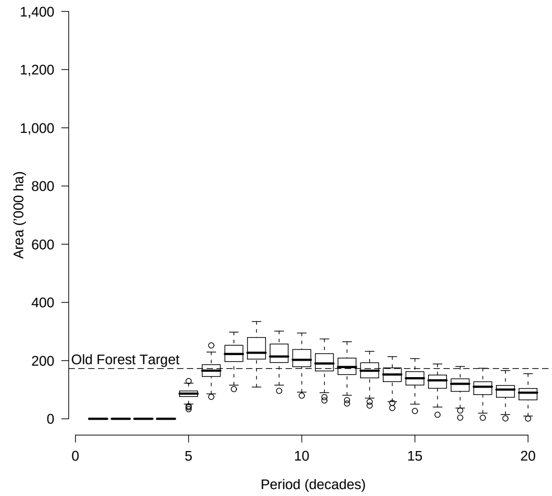



(d) 25% of Baseline Harvest Request

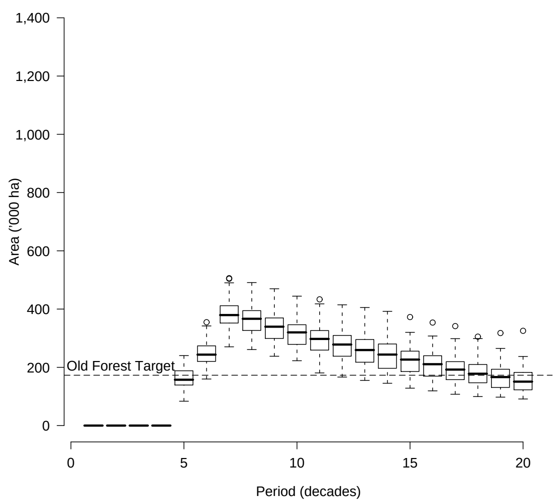
Figure 2.7: Harvest request achievement for deficit forest.



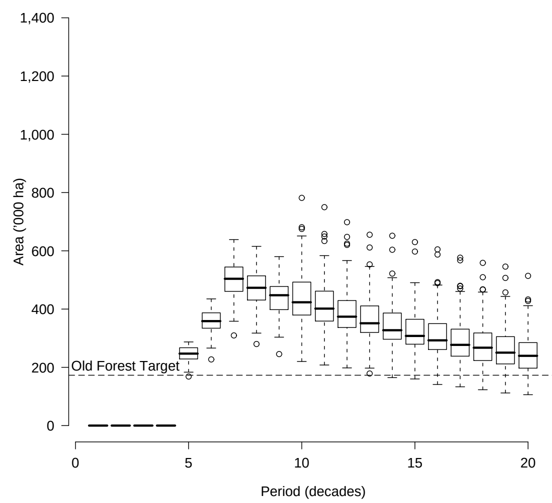

(a) 100% of Baseline Harvest Request




(b) 75% of Baseline Harvest Request




(c) 50% of Baseline Harvest Request




(d) 25% of Baseline Harvest Request

Figure 2.8: Old forest achievement for deficit forest.

constraint in order to determine a feasible solution. In the event that it could not, the harvest level constraint would drop incrementally until a solution was feasible.

Although improvement was clearly evident, the old forest indicator did not respond as positively to the harvest level reduction. Once again, this would be partly related to the model formulation. Even though the modeled objective function attempted to find a solution that maximized the minimum amount of old forest the solution was still subject to the harvest level constraint. Secondly, this and other analyses (Sutherland *et al.*, 2004) suggest that seral stage distribution is especially sensitive to disturbance. Wildfire disturbance directly alters seral stage distribution on the landscape by resetting stand ages. In the case of timber supply the model can adapt somewhat to the disturbance by harvesting at different ages whereas the old forest is eliminated for at least 90 years and possibly longer if it is reset by disturbance again before maturity.

Although there was a large degree of variability, all three forests required a major harvest level reduction to significantly increase the probability of both indicator targets. Even with an initial abundance of old-forest, as in the case of the surplus forest, maintaining old-forest area into the future may require a substantial reduction in harvest levels. At 75% of the deterministic base case harvest level, the old-forest target is not achieved in more than 50% of the runs by period 18.

## 2.5 Conclusions

In contemporary forest management planning processes forest indicators are projected over short (20 years), medium (20-50 years) and long-term (up to 200 years) time spans in order to assess the outcome of various management actions. The management actions are largely linked to deterministic processes such as regular harvest intervals and intensities. Although it is often recognized that projecting indicators into the future based solely on deterministic processes is an approximation of reality, many forest estate modeling exercises fall short of explicitly identifying uncertainty in forecasts (Fall *et al.*, 2004).

The source of uncertainty in modeling projections is diverse but the random nature and considerable impact of wildfire tends to be a major contributor in Western Canada's boreal forest. Failing to account for this impact and adopting a deterministic approach can lead to misleading representations of forest dynamics (McCarthy and Burgman, 1995) and therefore flawed projections of many indicators such as old forest area.

The approach presented in this analysis provides one avenue to test the robustness of deterministic forest indicator projections under an unpredictable disturbance regime. The addition of the Monte Carlo simulation component provided a probability distribution of results rather than one single improbable projection. However, a probability distribution adds a new level of complexity for decision makers. "How does one define sustainability?" becomes a very pertinent question and likely one that will need to be answered on a case by case basis. Managers will need to determine stakeholder's willingness to accept risk and recognize the potential implications now and in the future for this decision. From our example it is evident that a completely risk averse approach would result in a susbstantial cost to the current generation. On the other hand, high risk behavior now could have considerable implications for future generations.

The results of this particular analysis may not necessarily conclude that forest values as projected in forest management plans are unsustainable. Many opportunities exist that could reduce the impact of wildfire on the sustainability of forest values which were not included in this project. For example, enhanced fire protection with particular focus on the use of shadow prices (Armstrong and Cumming, 2003), spatially defined old growth management areas that may or may not be part of the harvestable landbase and fire salvage among other strategies may be employed to reduce the risk of failing to achieve targets. Overall though, managers should have an indication of the range of uncertainties when assessing plans and a realization that it may not be possible to meet all economic, social and biodiversity goals at all times (Sutherland *et al.*, 2004).

The modeling approach described and demonstrated in this paper is an extension of an approach developed by Armstrong (2004). Opportunities to build upon this particular variation of the concept are evident. For example, rather than using a no-disturbance model as the deterministic base case, a model with a fixed annual rate of burn similar to that of Reed and Errico (1986) may be more suitable. The fixed rate could be based on the calculated mean of the 31 year fire regime dataset. This would result in the eventual attainment of a steady state in harvest volumes and old forest. The variation around this deterministic base case may provide evidence that modeling fire as a fixed rate of disturbance has limited utility.

A second initiative would be to apply the methodology on a real world example. Although the use of hypothetical forests was very applicable for simple demonstration purposes, the concept should be further refined using actual forest inventory data and yield tables. For the methodology to be used in an operational setting the model will need to increase in complexity to reflect the multiple-aspects of forest management problems and the intricacies of forestry data.

# Bibliography

Acton, D.F., Padbury G.A., and Stushnoff C.T. 1998. *The Ecoregions of Saskatchewan.* University of Regina. Canadian Plains Research Center. Regina, SK, Canada.

Andison, D. 2006. Determining island remnants and meso-scale fire patterns in saskatchewan – part 2: Island remnant patterns. Technical report. Saskatchewan Forest Centre. Prince Albert, SK, Canada. 40 p.

Armstrong, G.W. 1999. A stochastic characterisation of the natural disturbance regime of the boreal mixedwood forest with implications for sustainable forest management. *Can. J. Forest Res.* **29**(4), 424–433.

Armstrong, G.W. 2004. Sustainability of timber supply considering the risk of wildfire. *For. Sci.* **50**(5), 626–639.

Armstrong, G.W., and Cumming, S.G. 2003. Estimating the cost of land base changes due to wildfire using shadow prices. *For. Sci.* **49**(5), 719–730.

Boychuk, D., and Martell, D.L. 1996. A multistage stochastic programming model for sustainable forest-level timber supply under risk of fire. *For. Sci.* **42**(1), 10–26.

Boyland, M., Nelson J., and Bunnell F.L. 2005. A test for robustness in harvest scheduling models. *For. Ecol. Manage.* **207**(1-2), 121–132.

Bridge, S.R.J., Miyanishi, K., and Johnson, E.A. 2005. A critical evaluation of fire suppression effects in the boreal forest of Ontario. *For. Sci.* **51**(1), 41–50.

CCFM 2003. *Defining Sustainable Forest Management in Canada. Criteria and Indicators. 2003.*. Canadian Council of Forest Ministers.

CSA 2002. *Sustainable Forest Management: Requirements and Guidance.* number Z809-02 In: *CSA Standard.* Canadian Standards Association.

Cumming, S.G. 1997. Landscape Dynamics of the Boreal Mixedwood Forest. PhD thesis. University of British Columbia. Vancouver, B. C., Canada.

Cumming, S.G. 2005. Effective fire suppression in boreal forests. *Can. J. For. Res* **35**(4), 772–786.

Dykstra, D.P. 1984. *Mathematical Programming for Natural Resource Management.* McGraw-Hill Series in Forest Resources. McGraw-Hill Book Company. New York. 318 p.

Eberhart, K.E., and Woodard, P.M. 1987. Distribution of residual vegetation associated with large fires in alberta. *Can. J. Forest Res.* **17**(10), 1207–1212.

Fall, A., Fortin, M.J., Kneeshaw, D.D., Yamasaki, S. H., Messier, C., Bouthillier, L., and Smyth, C. 2004. Consequences of various landscape-scale ecosystem management strategies and fire cycles on age-class structure and harvest in boreal forests. *Can. J. Forest Res.* **34**(2), 310–322.

FSC Canada 2004. *National Boreal Standard.* Forest Stewardship Council. Canada Working Group.

Gassman, H.I. 1989. Optimal harvest of a forest in the presence of uncertainty. *Can. J. For. Res* **19**, 1267–1274.

Johnson, E.A., Miyanishi, K., and O'Brien, N. 1999. Long-term reconstruction of the fire season in the mixedwood boreal forest of Western Canada. *Can. J. Bot.* **77**(8), 1185–1188.

Johnson, K.N., and Scheurman, H.L. 1977. *Techniques for Prescribing Optimal Timber Harvest and Investment under Different Objectives – Discussion and Synthesis.* number 18 In: *Forest Science Monograph.* Society of American Foresters.

Kachmar, M., and Sanchez-Azofeifa, G.A. 2006. Detection of post-fire residuals using high- and medium-resolution satellite imagery. *For. Chron.* **82**(2), 177–186.

Kawaguchi, T., and Maruyama, Y. 1976. A note on minimax (maximin) programming. *Management Sci.* **22**, 670–676.

Klenner, W., Kurz, W., and Beukema, S. 2000. Habitat patterns in forested landscapes: management practices and the uncertainty associated with natural disturbances. *Comput. Electron. Agric.* **27**(1-3), 243–262.

Martell, D.L. 1980. The optimal rotation of a flammable forest stand. **10**(1), 30–34.

McCarthy, M.A., and Burgman, M.A. 1995. Coping with uncertainty in forest wildlife planning. *For. Ecol. Manage.* **74**(1-3), 23–36.

Miyanishi, K., and Johnson, E.A. 2001. Comment - A re-examination of the effects of fire suppression in the boreal forest. *Can. J. Forest Res.* **31**(8), 1462–1466.

MOSEK 2009. MOSEK optimization tools.

Murphy, P.J. 1985. Methods for Evaluating the Effects of Forest Fire Management in Alberta. Ph.D. dissertation. Department of Forestry. University of British Columbia. Vancouver.

Naelapea, O., and Nickeson, J. 1998. SERM forest fire chronology of Saskatchewan in vector format.. Data set. Available on-line [http://www.daac.ornl.gov] from Oak Ridge National Laboratory Distributed Active Archive Center.

Parisien, M.A., Hirsch, K.G., Lavoie, S.G., Todd, J.B., and Kafka, V.G. 2004. Saskatchewan fire regime analysis. Inf. Rep. NOR-X-394. Northern Forestry Centre. Canadian Forest Service. Natural Resources Canada. Edmonton, AB, Canada. 49 p.

Peter, B., and Nelson, J. 2005. Estimating harvest schedules and profitability under the risk of fire disturbance. *Can. J. Forest Res.* **35**(6), 1378–1388.

Reed, W.J., and Errico, D. 1986. Optimal harvest scheduling at the forest level in the presence of risk of fire. *Can. J. Forest Res.* **16**, 266–278.

Reed, W.J. 1984. The effects of the risk of fire on the optimal rotation of a forest. *J. Environ. Econ. Management* **11**(2), 180–190.

Remsoft 2009. Woodstock.

Rock, T. 1996. A fire history in saskatchewan. Wildlife Branch. Saskatchewan Environment and Resource Management. Prince Albert, SK, Canada. 37 p.

Routledge, R.D. 1980. The effect of potential catastrophic mortality and other unpredictable events on optimal forest rotation policy. *For. Sci.* **26**(3), 389–399.

Sutherland, G.D., Eng, M., and Fall, S.A. 2004. Effects of uncertainties about stand-replacing natural disturbances on forest-management projections. *B.C. J. Ecosys. Manage* **4**(2), 53–70.

The Montréal Process 1999. *Criteria and Indicators for the Conservation and Sustainable Management of Termperate and Boreal Forests*. 2nd ed.

Turner, M.G., and Romme, W.H. 1994. Landscape dynamics in crown fire ecosystems. *Landscape Ecology* **9**(1), 59–77.

Van Wagner, C.E. 1983. Simulating the effect of forest fire on long-term annual timber supply. *Can. J. Forest Res.* **13**, 451–457.

Walker, H.V., Ross-Walker, M., and Clive, J. 1996. *Historical Review of the Economic Impacts of Changing Resource Use on Communities in the Prince Albert Model Forest Region*. Prince Albert Model Forest Association. Prince Albert, SK, Canada.

# Chapter 3

# An economic evaluation of a standing timber buffer stock strategy

## 3.1 Introduction

Stand replacing disturbance plays a major role in shaping the state of the boreal forest. For example, managing forests for timber harvest is known to have a significant impact on the age class distribution (Gustafson and Crow, 1998). Harvest, of course, is not the only form of disturbance on the boreal landscape; there are a number of stand replacing elements. In Western Canada fire is a major disturbance (Parisien *et al.*, 2004). Unlike harvest, which occurs at relatively regular intervals and fixed intensities, fire disturbance is random and can be characterized as stochastic (Armstrong, 1999). Difficulty arises when attempting to incorporate the impact of stochastic fire disturbance in forest management plans. The considerable variation within the fire regime make it difficult to predict the actual outcome of any particular forest estate model scenario, even if the deterministic harvest schedule is followed precisely. This uncertainty contributes to the risk of failure in the achievement of indicator targets. A continuous cycle of re-calibration through the use of periodic replanning helps to reduce this risk of long term deviation from sustainability. However, replanning does not ensure stability of indicators modeled, nor does it improve our present day projections of the future forest state.

In Canada, two major strategies have evolved to cope with this uncertainty. The first strategy is based on the theory that a sufficient buffer stock of standing merchantable timber will provide stability in timber harvest levels. In order to build a buffer stock of timber the harvest rate is immediately set below the mathematically optimal level calculated without the consideration of fire. A series of papers have supported this notion. Using a simulation model and a hypothetical forest Van Wagner (1983) illustrated how a reduced harvest level can stabilize long term timber flow. Although Van Wagner's progressive approach entailed randomly selecting units within his 1,000 unit forest, the amount of area burned in each simulation occurred at a constant rate. Reed and Errico (1986) supported Van Wagner's conclusions by using an variation of the Johnson and Scheurman (1977) linear programming model II formulation. In Reed and Errico's model an amount of area equal to the mean rate of disturbance is burned each period. Similar to Van Wagner, the model assumed an equal probability of burning regardless of stand age. The resulting harvest level would therefore be reduced according to the impact of the mean annual fire rate. If the resulting harvest schedule was followed, a buffer stock of timber would be left to compensate for the actual fire loss. Although their approach dealt with fire in a deterministic fashion, they further suggested that a cycle of implementing their fire model solution for a period, updating the inventory to account for the actual area that burned and then resolving the model would provide a reasonable approximation of the impact of stochastic fire disturbance. Boychuk and Martell (1996) built on the Reed and Errico approach and developed a model which incorporated fire as a stochastic process. Their stochastic model represented fire as a draw from a discrete two point probability distribution resulting in the possibility of a high fire period or a low fire period. Their analysis again concluded that a buffer stock of timber was necessary to compensate for potential fire losses. Further, they also suggested that an immediate reduction in the harvest level may actually result in an increase in the average harvest volume over the planning horizon.

The buffer stock strategy essentially advocates an immediate harvest level reduction regardless of the actual amount of fire disturbance that will occur during the upcoming planning period. A second strategy to cope with the uncertainty posed by fire is to simply incorporate the impact of fire after it has occured. We will refer to this as the no buffer stock strategy. This approach assumes the impact of the area disturbed in the previous period will be realized by resolving the model with an updated inventory at the start of the subsequent planning period. Under this strategy the possibility of unduly forgoing timber harvest opportunity in the short term future is eliminated. Using a simulation/optimization modeling system the probable results of each of these strategies is explored for a range of indicators with the intent of reflecting forest management planning in Canada.

## Modeled Indicators

Contemporary forest management plans are multidimensional in nature. Although timber often remains a major driver it is no longer the sole indicator in a forest estate model. The public also demands that ecological values be maintained. The maintenance of biodiversity is an important component of ecological integrity and is recognized as a criterion for sustainable forest management as defined by the Montreal process and adopted by the Canadian Council of Forest Ministers (The Montréal Process, 1999; CCFM, 2003). Our approach in this paper includes an indicator for both timber harvest and biodiversity (old forest area).

It is common for the economic component of a forest estate model to be reflected in the objective function of a linear programming based model. Likely the most common approach is to maximize financial benefit indirectly through timber production or directly through net revenue. Perhaps not as common in Canada is an objective function that maximizes net present value. Net present value offers an alternative method of evaluating different investment decisions. In forest management the investment decision could be represented by a management plan scenario or a particular policy. The net present value of each alternative can be calculated and the estimated returns could be compared. In our case the investment choice or policy we will be exploring is the buffer or no buffer strategies.

Using net present value as a decision criterion, multiple draws of a simulation/optimization modeling system is used to compare the estimated returns of two different strategies (buffer or no buffer). The probability of achieving the key indicator values such as timber harvest and old forest are further examined.

## 3.2 Modeling System

### Model Overview

The system of models used for this study consists of an optimizing forest estate model used to choose optimal timber harvest plans, nested within a Monte Carlo model of stand-replacing forest fires. The model loops through a number of draws representing possible realizations of the fire regime projected over time. Each draw represents a sequence of twenty decades comprising the simulation horizon.

Outputs from the model include the sequence of harvest and old forest levels achieved and the net present value for each particular draw. The results for mulitple draws are then plotted to create a distribution of probable harvest levels, old forest levels and net present values. Separate runs of the model are completed representing the buffer stock and no buffer stock strategies. The indicator distributions for the buffer stock and no buffer stock strategies are then compared to investigate the implications of either policy choice.

The optimization model used in this system is sometimes referred to as a model III formulation (Martell, 1994). A model III formulation allows for the incorporation of deterministic fire disturbance. Typically the deterministic fire disturbance rate (refered to in the model formulation as the burn fraction) reflects the mean annual fire rate for the landbase being modeled. A model III follows the negative exponential survivorship theory (Van Wagner, 1978; Johnson and Gutsell, 1994) with each age class having equal probality of burning. The model implements the fire disturbance by periodically reseting the birth period of a proportion (equal to the burn fraction) of each age class in the forest.

A schematic description of the modeling system is provided in Fig. 3.1. The modeling system solves the optimization model twice for every planning period. The model first solves with the deterministic burn rate set to "Y" (reflecting the mean annual disturbance rate). The burn rate is then set to zero and the previous harvest level result is used as a maximum harvest level constraint in the second optimization. The harvest solution from

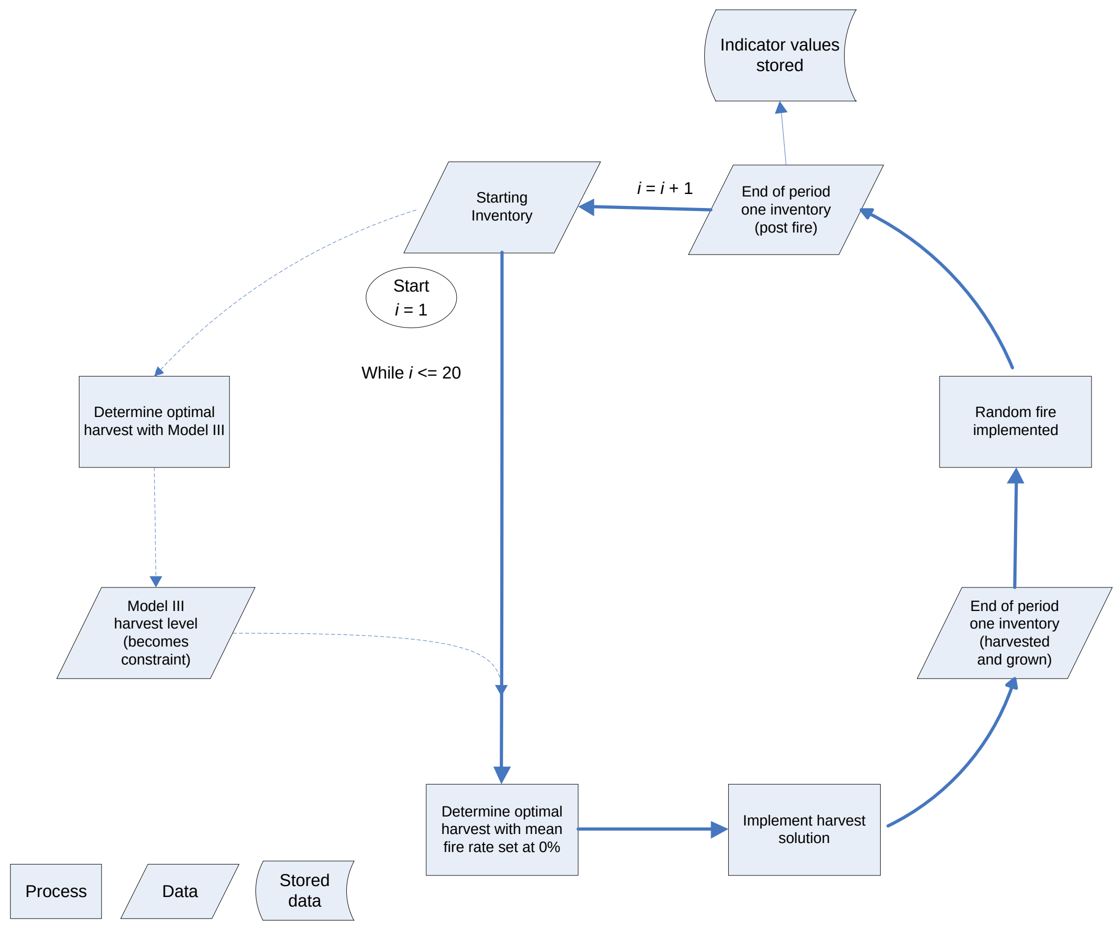



Figure 3.1: Overview of Chapter 3 Modelling System

the second optimization is implemented in the forest. At this point the first period harvest level and net present value are stored. The forest is then updated for growth and random fire disturbance. The resulting old forest level is stored for that period and the cycle loops back.

The net revenue stored for each period is the first period net revenue and therefore has been discounted as if it were the first planning period in the horizon. In actuality it may not be the first period and the stored net revenue values were discounted according to the actual planning period after a series of model draws were complete. An assumption was made that all activities and revenues occur at the mid-point of the planning period (year 5) and therefore all discounting reflects this.

To obtain results for the buffer stock strategy, "Y" is set to the mean annual burn rate. Thus the implemented harvest will be at a level that accounts for the mean rate of fire disturbance. This will mimic harvest planning that incorporates an immediate reduction while allowing the "actual" random fire disturbance to occur in the forest. To obtain results for a no buffer stock strategy, "Y" is set to zero for the entire program loop. The harvest level constraint will then have no effect and the implemented solution will reflect planning without a harvest level reduction.

### Monte Carlo burn proportion model

The fire regime is described as a sequence of projected annual burn proportions which are drawn randomly from the fire history. Because the time step in the simulation loop is one decade, these annual projections of burn proportions need to be converted into decadal proportions. The burn rate for each decade is calculated as

$$\Lambda = 1 - \prod_{i=n}^{n}(1 - \lambda_i) \tag{3.1}$$

where $\lambda_i$ is the randomly drawn annual burn rate. The decadal burn rate, $\Lambda$, is the proportion of the forest area burned at least once in the decade.

### Model Formulation

The formulation was built using the Woodstock forest modeling software (Remsoft, 2009). Mosek Optimization Tools (MOSEK, 2009) was used to determine a solution. The model is described in formal notation as follows.

The objective function maximizes the net present value (present value of the revenue minus present value of the costs) over the planning horizon comprising $H$ periods.

$$maxZ = \sum_{j=1}^{H} PVR_j - \sum_{j=1}^{H} PVC_j \tag{3.2}$$

Where the present value of the revenue (PVR) is calculated as:

$$PVR_j = \sum_{i=-M}^{j-1} R_{ij} x_{ij} \tag{3.3}$$

As in a model II formulation, in the model III formulation it is necessary to track "birth" and "harvest" periods for stands in the forest. The birth period for the oldest existing stand in the forest is period $-M$. The birth period for a one-period old stand at the beginning of the planning horizon is period 0. The decision variables $x_{ij}$ represents the area of forest born in period $i$ harvested in period $j$. The discounted revenue for harvesting forest born in period $i$ in period $j$ is represented by $R_{ij}$. The age at harvest (in periods) can be calculated as $j - i$.

The discounted revenue is calculated with the assumption that all revenue is accrued at the mid-point of the planning period (year 5). Discounted revenue is calculated as:

$$R_{ij} = \frac{v_{ij} p}{(1+I)^{(j*10-5)}} \tag{3.4}$$

Where $v_{ij}$ is the yield table volume for the forest representing the volume per hectare obtained from a stand which is born in period $i$ and harvested in period $j$. $p$ is the price paid for one m$^3$ of timber. $I$ represents the discount rate.

Where the present value of the costs (PVC) is calculated as:

$$PVC_j = \sum_{i=-M}^{j-1} C_j x_{ij} \tag{3.5}$$

The discounted cost is calculated as:

$$C_j = \frac{(h+s)}{(1+I)^{(j*10-5)}} \tag{3.6}$$

Where $h$ is the cost of harvesting and $s$ is the cost of silviculture for one unit of forest. An assumption is made that both harvest and silviculture costs are static regardless of birth period.

An even flow constraint was included which ensures that the harvest volume in each period equals that of the previous period.

$$F_k - \sum_{i=-M}^{k} v_{ik} x_{ik} = 0 \qquad k = 1, 2, \ldots, H \tag{3.7}$$

$$F_k - F_{k-1} = 0 \qquad k = 1, 2, \ldots, H \tag{3.8}$$

A pair of constraints were used to calculate the age class distribution in each period, after harvest, but before growth. $D_{jk}$ represent the area of forest in age class $j$ in period $k$. Eq. 3.9 summarizes the area harvested in each period or, equivalently, the area of forest zero periods old, in each period of the planning horizon. Eq. 3.10 calculates the area in all the other age classes.

$$D_{0k} - \sum_{j=-M}^{k} x_{ik} = 0 \qquad k = 1, 2, \ldots, H \tag{3.9}$$

$$D_{k-i,k} - \sum_{i=-M}^{k} \sum_{j=k+1}^{H} x_{ij} + w_i = 0 \qquad k = 1, 2, \ldots, H \tag{3.10}$$

The variable $E_k$ is used to represent the area of old growth in period $k$. In Eq. 3.11, the constant $U$ represents the youngest age class considered to be old growth. Eq. 3.12 is the constraint used to ensure that old growth area in periods exceeds the desired threshold, $\bar{E}$.

$$E_k - \sum_{e=U}^{H+M} D_{ek} = 0 \qquad k = 1, 2, \ldots, H \tag{3.11}$$

$$E_k \geq \bar{E} \qquad k = s, s+1, \ldots, H \tag{3.12}$$

In order to ensure that the model does not liquidate the forest at the end of the planning horizon, we implement constraints that ensure that the volume of growing stock is non-declining over a period at the end of the planning horizon. Eq. 3.13 summarizes the growing stock volume, $G_k$, in each period $k$. Eq. 3.14 is non-declining from period $Q$ to the end of the planning horizon, $H$.

$$G_k - \sum_{j=k}^{H} x_{ij} v_{ik} + w_i v_{ik} = 0 \qquad k = 1, 2, \ldots, H \tag{3.13}$$

$$G_k - G_{k-1} \geq 0 \qquad k = Q, \ldots, H \tag{3.14}$$

A model III formulation implemented in the Woodstock modeling platform requires a series of constraints not typical of a model II. In the model III variety described here, fire is incorporated through the use of an "action". What we will refer to as the "pre-burn" action is actually a decision variable in the model. For each hectare of forest that is selected to be pre-burnt, a proportion equal to the burn rate is re-born (transitioned to age zero). A proportion equal to 1 - burn rate maintains its current birth period. No yields are gained from the implementation of this decision variable. Thus, in order for the model to implement the pre-burn, a constraint is required. The constraint forces all forest area in the model to be selected for pre-burning in every period. By doing so, a proportion of the entire forest area, equal to the burn rate, is re-born in every period (disturbed by fire). The constraint that forces the entire forest to be selected for the pre-burning could be expressed as follows:

$$b_k = A_k \tag{3.15}$$

Where $b_k$ is the total amount of area selected to undergo pre-burning in period $k$ and $A_k$ is the total amount of forest area in period $k$. $b_k$ is calculated as:

$$b_k - \sum_{i=-M}^{k} b_{ik} = 0 \qquad k = 1, 2, \ldots, H \tag{3.16}$$

In this variation of a model III formulation area is tracked as to whether or not it has been selected for pre-burning regardless of birth period. For example, in period two of the planning horizon there will be two "bins" for each birth period existing. One bin will be the original inventory area while the other will be area that was selected for pre-burning in the previous period (period one). In period three there will be three bins for each existing birth period. One will be for the original inventory, one will be for area that was selected for pre-burning in period one and one will be for area that was selected for pre-burning in the previous period (period two). As a result of our constraint forcing all area to be selected for pre-burning, all bins will be empty with the exception of the set originating from the previous period. $A_k$ (the total amount of area in period $k$) reflects this tracking system and is calculated as:

$$A_k - \sum_{i=-M}^{k} \sum_{j=k+1}^{H} x_{i_o j} + x_{i_{b_{\leq k}} j} + b_{i_o j} + b_{i_{b_{\leq k}} j} + w_i = 0 \qquad k = 1, 2, \ldots, H \tag{3.17}$$

Where $x_{i_o j}$ is the area harvested of birth period $i$ in period $j$ which is of the original inventory origin. $x_{i_{b_{\leq k}} j}$ is the area harvested of birth period $i$ in period $j$ which is of origin from pre-burning in period $k$ or earlier. Similarly, $b_{i_o j}$ is the area pre-burned of birth period $i$ in period $j$ which is of the original inventory origin. $b_{i_{b_{\leq k}} j}$ is the area pre-burned of birth period $i$ in period $j$ which is of origin from pre-burning in period $k$ or earlier. $w_i$ is the area of birth period $i$, of either origin, that has not been harvested or pre-burned.

A series of area constraints are required in this model III variation. First of all, a starting inventory constraint is needed to ensure that all of the area in each of the initial birth periods is assigned to a valid harvest or no harvest decision.

$$\sum_{j=1}^{H} x_{i_o j} + b_{i_o j} + w_{i_o} = A_i \qquad i = -M, -M+1, \ldots, 0 \tag{3.18}$$

Next, two area transfer constraints are needed to ensure area is re-born after disturbance. The first constraint ensures that area is reborn after harvest and fire ("fire" being the proportion of area selected for pre-burning that is reborn).

$$b_{jj} + \sum_{k=j+1}^{H} x_{j_f k} + b_{j_f k} + w_{j_f} - \sum_{i=-M}^{j-1} x_{ij} + (BF)(b_{ij}) = 0 \quad j = 1, 2, \ldots, H \tag{3.19}$$

$b_{jj}$ is the amount of area of birth period $j$ that is subject to pre-burning in the same period as its birth (period $j$). This area was already reborn due to timber harvest earlier in the period and now has the potential to be disturbed for a second time in the same period. Due to our constraint forcing all area to be selected for pre-burning this bin will hold all area that was harvested in period $j$. $x_{j_f k}$ is the amount of area of birth period $j$, of a fire origin, that is harvested in period $k$. Similarily, $b_{j_f k}$ is the amount of area of birth period $j$, of a fire origin, that is selected for the pre-burn decision variable in period $k$. $w_{jf}$ is area of birth period $j$, of a fire origin, that is neither harvested nor selected for the pre-burn in the remainder of the planning horizon. $x_{ij}$ is area of birth period $i$, of any origin, harvested in period $j$. $BF$ is the burn fraction and $b_{ij}$ is area of birth period $i$ selected for pre-burning in period $j$.

Another area transfer constraint ensures that area represented by $b_{jj}$ in the previous equation is reborn.

$$\sum_{k=j+1}^{H} x_{j_{hf}k} + b_{j_{hf}k} + w_{j_{hf}} - \; b_{jj} = 0 \tag{3.20}$$

$x_{j_{hf}k}$ is the area of birth period $j$, of an origin of both harvest then pre-burning in the same period, that is harvested in period $k$. Again, $b_{j_{hf}k}$ is the area of birth period $j$, of an origin of both harvest then pre-burning in the same period, that is subject to pre-burning in period $k$. $w_{jhf}$ is area of the same origin that is neither harvested nor pre-burned in the remainder of the planning horizon.

The last area constraint ensures that a proportion of pre-burn area equal to 1 - the burn fraction is transitioned to another bin of the same birth period.

$$\sum_{k=j+1}^{H} x_{i_{b_i}k} + b_{i_{b_i}k} + w_{i_{b_i}} - b_{ij}(1 - BF) = 0 \quad i = -M, -M+1, \ldots, 0 \tag{3.21}$$

$x_{i_{b_j}k}$ represents area of birth period $i$, of a pre-burn origin of period $i$, that is harvested in period $k$. $b_{i_{b_i}k}$ is area of birth period $i$, of a pre-burn origin of period $i$, that is pre-burned in period $k$. $w_{i_b}$ is area that is not harvested nor pre-burned for the remainder of the planning horizon but is of a pre-burn origin of period $i$. $b_{ij}$ is the area of birth period $i$ pre-burned in period $j$ and $BF$ is the burn fraction.

## 3.3 Model Application

To demonstrate the methodology described in this paper a model was developed using the productive forest area and fire regime from the Prince Albert Forest Management Agreement (PA FMA) area in Saskatchewan, Canada.

### Input Data

The input data for this analysis was very similar to Chapter 2. For this study, two different variations of a hypothetical single species forest representing a normal and surplus starting inventory age class distribution were used. This area was split equally into age classes according to Table 3.1.

Table 3.1: Alternative initial age class distributions for normal and surplus forests.

| Age Class (yr) | Normal (% area) | Surplus (% area) |
|---|---|---|
| 10 | 10 | 0 |
| 20 | 10 | 0 |
| 30 | 10 | 0 |
| 40 | 10 | 0 |
| 50 | 10 | 0 |
| 60 | 10 | 10 |
| 70 | 10 | 10 |
| 80 | 10 | 10 |
| 90 | 10 | 10 |
| 100 | 10 | 10 |
| 110 | 0 | 10 |
| 120 | 0 | 10 |
| 130 | 0 | 10 |
| 140 | 0 | 10 |
| 150 | 0 | 10 |

This study used the identical Chapman-Richards growth function to determine yields as that used by Armstrong (2004), $y(t) = b_1 \left(1 - e^{b_2 t}\right)^{b_3}$, where $t$ is age in years, $y$ is the timber yield (m$^3$ ha$^{-1}$), and $b_1$, $b_2$ and $b_3$ are coefficients with the values 200, -0.026604 and 5 respectively. These coefficients were chosen such that the maximum yield is 200 m$^3$ ha$^{-1}$ and the mean annual increment (MAI) reaches a maximum at 100 years of age. The maximum MAI for this yield curve is 1.392 m$^3$ ha$^{-1}$ yr$^{-1}$.

## Fire Regime Model

The fire regime used in this exercise is derived directly from the actual fire history of the PA FMA area through the random selection of burn years in a thirty-one year area burned dataset. The dataset is based on the fire history of the PA FMA area during the 1977 to 2007 period. A complete description of the methodology used to create the area burned dataset is provided in Chapter 2.

## Financial Parameters

To account for the potential sensitivity of the model results to the discount rate choosen, a range of figures were used. For both hypothetical forests the model was run using discount rates of 2.5%, 5% and 7.5%. The financial parameters in the model were theoretical in nature and do not represent empirical data. The intent of the figures used was to provide a general approximation of the average costs and revenues experienced by wood users in the boreal forest of western Canada. The value of timber was set at \$50/m$^3$, the harvest cost was \$5000/ha and the renewal cost was \$500/ha.

## 3.4 Results

The harvest and old forest level results for each strategy are illustrated in Fig. 3.2 and Fig. 3.3. Fig. 3.2 represents the normal forest while Fig. 3.3 represents the surplus forest, both at a 5% discount rate. The median, upper and lower quartiles are represented by the box. The whiskers represent the largest or smallest non-outlier value and the circles represent outliers.

The harvest level graphs (subfigures a and b) display the distribution of harvest levels that resulted from 100 draws of the system. The dashed line represents the initial even-flow harvest level that resulted from solving the optimization model oustide of the modeling system. Subfigures c and d represent the distribution of old forest levels achieved. The dashed line indicates the 10% old forest area target that was included as a constraint in the model. Regardless of the discount rate used, the same general trend was evident for both the harvest and old forest levels. For the sake of brevity, only the 5% discount rate is presented for either forest.

The second set of two box plots (Fig. 3.4 and Fig. 3.5) provide the distribution of net present values and cumulative harvest levels that were achieved for both the no buffer and buffer strategies. Fig. 3.4 represents the normal forest while Fig. 3.5 represents the surplus forest, both at a 5% discount rate. The net present value graph (subfigure a) provides an indication of the range of discounted financial return each strategy may produce. The total harvest level graph (subfigure b) provides the range of cumulative harvest levels over the entire 200 years modeled.

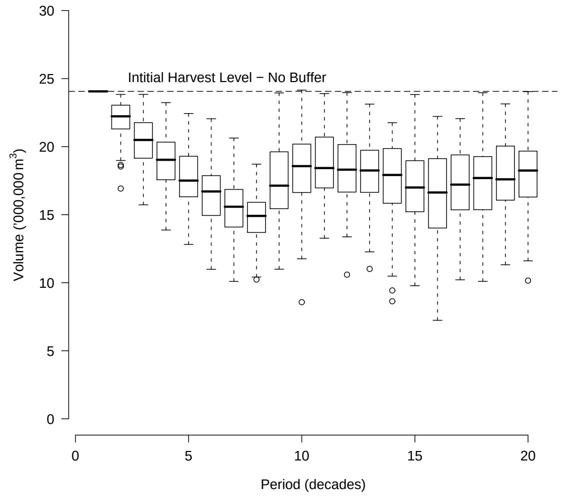
In general, the distribution of net present values were higher for the no buffer strategy on both forests and for all discount rates tested. The one exception to this would be the normal forest at a 2.5% discount rate. This forest/discount rate combination resulted in a higher return for the buffer strategy.

Although NPV tended to be higher in the absence of a buffer, it should be noted that the variance also tended to be higher. The range of potential NPV values was noticably wider as indicated by the boxplots. Table 3.2 and Table 3.3 provide the median, upper and lower quartile net present values for each discount rate modeled.

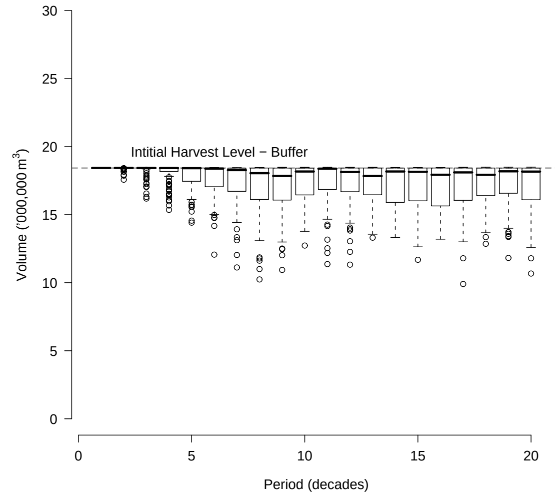
The 200 year cumulative harvest level distribution was higher for the no buffer stratey on all forest/discount rate combinations. This does not necessarily indicate that the harvest level will be higher at any given point in time for a no buffer strategy. It does however, indicate that there is a greater probability that a higher 200 year cumulative harvest volume will be achieved from the no buffer strategy.

## 3.5 Discussion

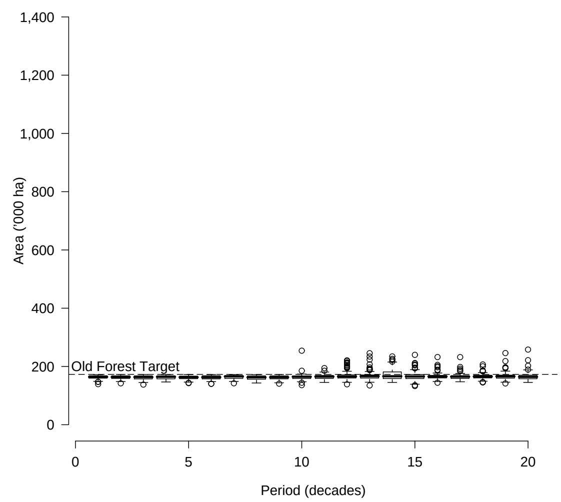
With respect to the normal forest, harvest levels proved to be variable in the absence of a timber buffer. At a 5% discount rate the system indicated a possible low of less than



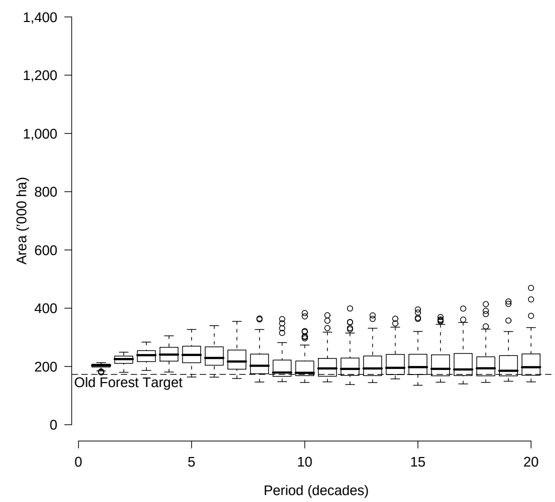

(a) No Buffer Harvest Levels




(b) Buffered Harvest Levels




(c) No Buffer Old Forest

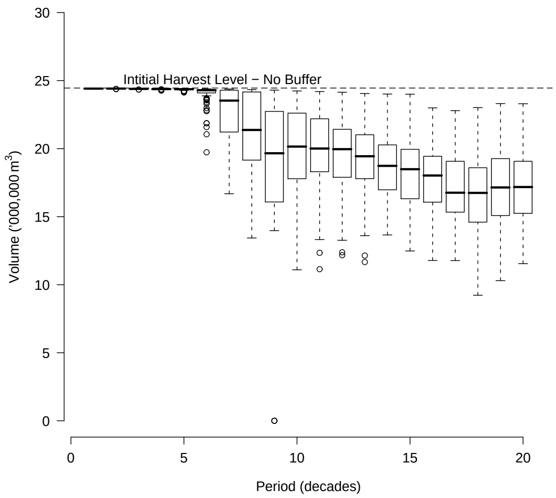



(d) Buffered Old Forest

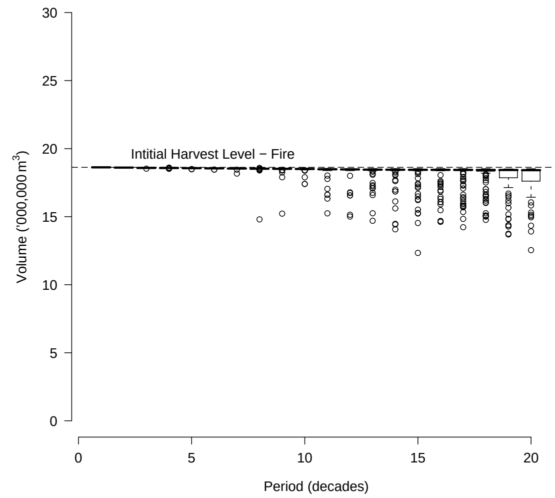
Figure 3.2: Harvest and old forest results for the no buffer and buffer strategies (Normal forest at a 5% discount rate)

Table 3.2: Distribution of net present values ($10^6$ \$) for discount rates for the normal forest.

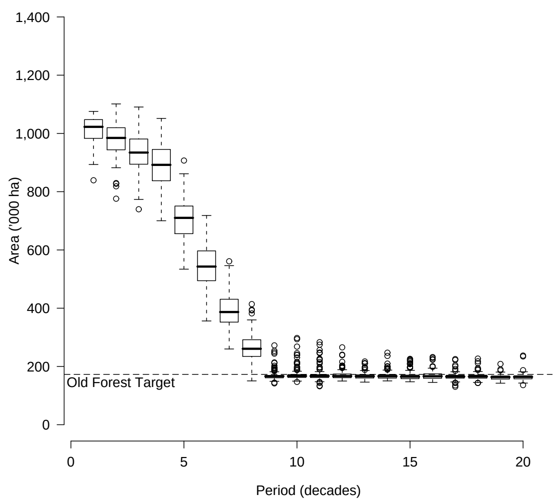
| | 2.5% No Buffer | 2.5% Buffer | 5% No Buffer | 5% Buffer | 7.5% No Buffer | 7.5% Buffer |
|---|---|---|---|---|---|---|
| Lower Quartile | 810 | 839 | 436 | 425 | 306 | 279 |
| Median | 836 | 868 | 455 | 434 | 315 | 281 |
| Upper Quartile | 880 | 895 | 466 | 439 | 319 | 283 |



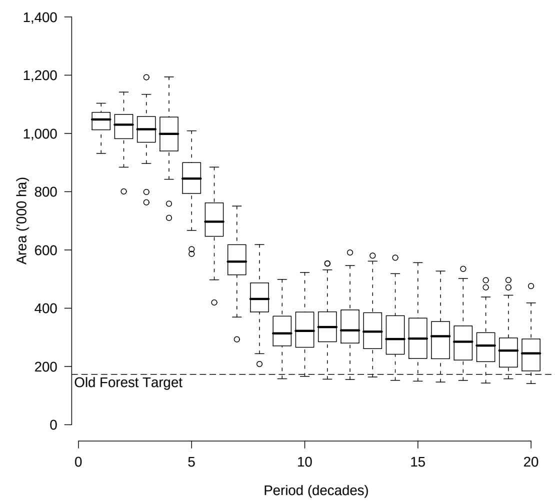

(a) No Buffer Harvest Levels




(b) Buffered Harvest Levels




(c) No Buffer Old Forest

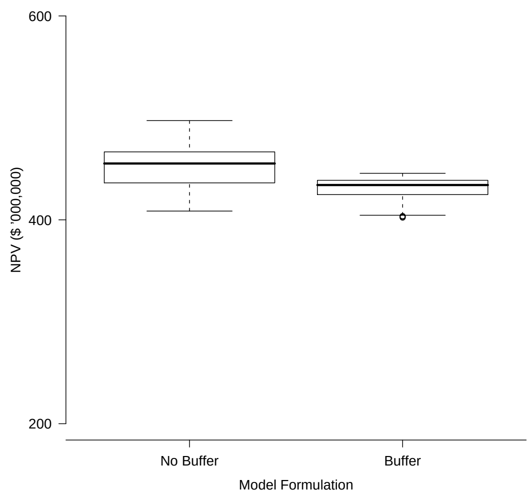



(d) Buffered Old Forest

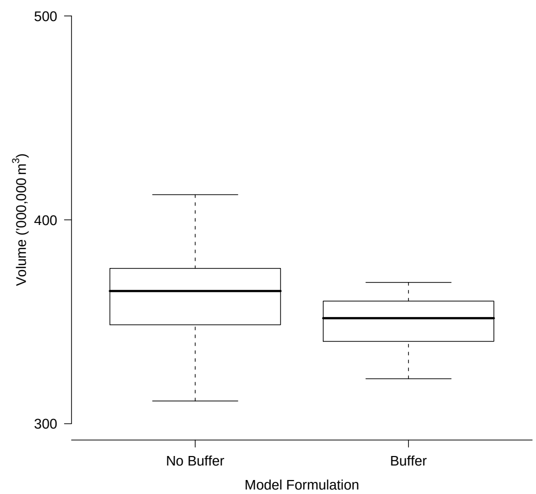
Figure 3.3: Harvest and old forest results for the no buffer and buffer strategies (Surplus forest at a 5% discount rate)

Table 3.3: Distribution of net present values ($10^6$ \$) for discount rates for the surplus forest.

| | 2.5% No Buffer | 2.5% Buffer | 5% No Buffer | 5% Buffer | 7.5% No Buffer | 7.5% Buffer |
|---|---|---|---|---|---|---|
| Lower Quartile | 1,792 | 1,484 | 974 | 762 | 657 | 507 |
| Median | 1,830 | 1,493 | 978 | 763 | 658 | 507 |
| Upper Quartile | 1,853 | 1,500 | 984 | 764 | 659 | 507 |

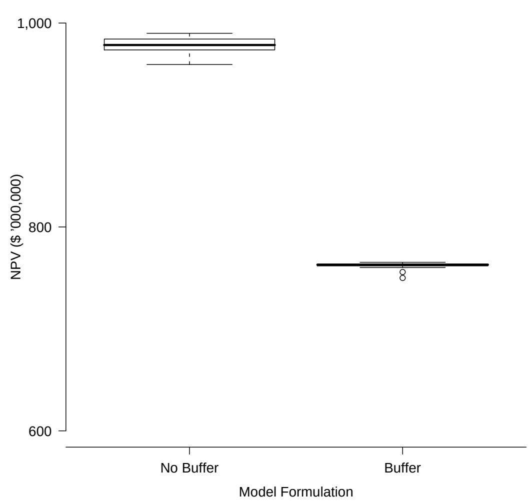



(a) Net Present Values

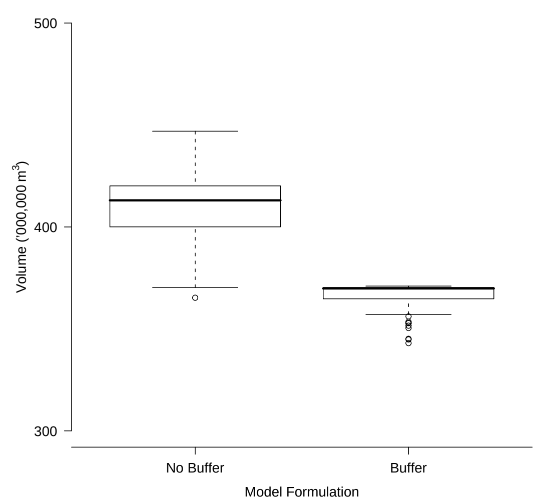



(b) Harvest Levels

Figure 3.4: NPV and cumulative 200 year harvest level results for the no buffer and buffer strategies (Normal forest at a 5% discount rate)




(a) Net Present Values




(b) Harvest Levels

Figure 3.5: NPV and cumulative 200 year harvest level results for the no buffer and buffer strategies (Surplus forest at a 5% discount rate)

8,000,000 m$^3$/period and a high of 24,062,000 m$^3$/period for the 100 model draws. However, these extremes were unlikely to occur and resulted from an improbable sequence of severe fire years or minor fire years respectively. Perhaps unexpected was the variability in harvest levels with the inclusion of a timber buffer. At a 5% discount rate the system indicated a possible low of less than 10,000,000 m$^3$/ period and a high of 18,431,000 m$^3$/period. This distribution of values indicate that the buffer stock provided through a model III formulation was not sufficient to stabilize harvest levels.

Continuing with the normal forest, the immediate harvest level reduction for the buffer strategy was 23.4% relative to the no buffer strategy. For 30 years into the planning horizon it is highly probable that the no buffer strategy continues to yield a significantly higher volume. Although the no buffer harvest appears likely to drop below the buffered harvest for particular periods in the horizon (periods 6-8 inclusive), much of the remaining periods have comparable upper and lower quartile distributions. In fact, there is a reasonable possibility that the no buffer harvest will be higher than the buffered harvest for much of the planning horizon. In the case of the normal forest the initial harvest cost of a buffer stock strategy is high yet it does not appear to effectively stabilize harvest levels nor significantly reduce the risk of a castrophically low harvest.

Due to an over abundance of standing merchantable growing stock at the outset of the planning horizon, the surplus forest harvest distributions indicated a slightly different trend. In the absence of a timber buffer the harvest level remained stable for the first 60 years. Once the intitial over abundance of standing merchantable growing stock is liquidated a high degree of variability is evident. In contrast, the buffer strategy harvest remains stable throughout the majority of the planning horizon. By comparing the two distributions it is likely a greater harvest will be achieved with the no buffer strategy for the first 140 years modeled. For the remainder of the planning horizon the harvest distributions are somewhat comparable, the buffered harvest level remains within the upper and lower quartiles of the no buffer harvest. Under a surplus forest scenario the additional standing merchantable timber created by a buffer stock strategy is effective at stabilizing harvest levels. However, a large amount of harvest opportunity is forgone immediately and throughout the planning horizon with this strategy.

The cumulative 200-year harvest totals displayed a similar trend regardless of forest or discount rate. As previously discussed, the buffer strategy harvest levels may be higher for particular periods. In general though, the total harvested timber over the 200 years modeled is likely to be higher in the absence of a buffer (See Fig. 3.4(b) and Fig. 3.5(b)).

In the absence of a timber buffer, old forest levels in the normal forest remained relatively stable. However, the target level was seldomly achieved throughout the planning horizon. Most often old forest area fell slightly below the target. This result is a function of the modelling system used. Due to the profit maximizing nature of the objective function the model solution will only carry enough old forest area necessary to satisfy the constraint

(10% in our example). The Monte Carlo simulation burns the inventory after the optimal solution is implemented thus driving old forest slightly below the target level. In contrast to the no buffer results, there was a high probability of achieving the old forest target when a buffer stock of timber is present. Although the possibility of failure does exist, the upper and lower quartiles stay above the target line for the entire planning horizon.

Although the over abundance of old forest slowed the onset, a similar result is present with the surplus forest. As the growing stock is liquidated we see that the old forest levels drop slightly below the target in the absence of a buffer. When a buffer is present the probability of maintaining at least the target is much higher for the duration of the planning horizon.

In general, the probability of falling significantly short of the old forest target was low with both strategies. This could be explained by the fact that the old forest target was a constraint in the optimization model formulation. As a constraint the model solution must first satisfy the old forest target in order to be feasible.

From a financial perspective net present value (NPV) was examined as a decision criterion for strategy selection. For the normal forest, results indicate that the strategy with the greatest financial return is sensitive to discount rate. At 2.5%, the range of potential NPVs overlap considerably for both strategies. Depending on the combination of fire years drawn either strategy could provide a higher financial return. As identified in Table 3.2, the median NPV is higher for the buffer strategy at 2.5%. This indicates that the probability of a greater profit is slightly higher with a buffer stock when using a 2.5% discount rate. However, the opposite would be true for the 5 and 7.5% discount rates. As discount rate increases the no buffer strategy is clearly favoured based on a net present value criteria.

Further examining our economic parameters can provide some insight into these NPV results. At a 2.5% discount rate our economic rotation age for a stand rises to 110 years rather than 100 years at discount rates of 5% and 7.5%. A 110 year economic rotation age likely contributes to an initial harvest level that is approximately 10% lower relative to 100 years. The result is an initial no buffer harvest level that is closer to that resulting from the buffer strategy. This reduction in harvest level early in the planning horizon causes a considerable decrease in the NPV.

Net present value distributions clearly favoured the no buffer strategy for the surplus forest. At all discount rates the no buffer strategy was significantly more profitable. A considerable degree of financial benefit is potentially forgone by the inclusion of a buffer strategy when an overabundance of standing merchantable growing stock is already present in the starting inventory.

However, the variability of value distributions should also be considered. As stated earlier, the variability in values appeared to be higher without a buffer. This result could have bearing on the policy decision chosen and would need to be considered alongside the firm's aversion to risk. Without a buffer there appears to be greater uncertainty in financial return. This uncertainty may be unacceptable and the firm may choose to accept a potentially lower return as a trade off for increased certainty.

## 3.6 Conclusion

Based on the results of this simple model exercise the implementation of a timber buffer stock strategy is questionable on a number of fronts. From a harvest level perspective, a considerable economic opportunity is lost immediately. The probability of a return on this "investment" later in the planning horizon is uncertain for both the normal and surplus forests. Essentially, a large up front payment is being made based on the possibility of a small benefit far into the future. There is also a risk that this small benefit may never be realized.

Although old forest targets appear to have a slightly higher probability of being achieved with a buffer neither strategy resulted in gross underachievement of this target. Due mainly to the nature of the modelling system a slight underachievement of the old forest target is likely at some point in the planning horizon with the no buffer strategy.

From a net present value perspective, all forest/discount rate combinations, with the exception of one, had the same results. It was clearly more profitable to not include a buffer. The normal forest with a 2.5% discount rate was the exception to this. While the possbility still exists that a no buffer strategy will be more profitable, it is more probable that a buffer stock will result in a higher net present value. This would be due to the finanancial parameters used for the analysis. There is an apparent shift to a greater economic rotation age when a 2.5% discount rate is used. The greater economic rotation age results in a lower initial harvest level for the normal forest which in turn has a significant impact on net present value. All other forest/discount rate combinations clearly favoured no buffer.

# Bibliography


Armstrong, G.W. 1999. A stochastic characterisation of the natural disturbance regime of the boreal mixedwood forest with implications for sustainable forest management. *Can. J. Forest Res.* **29**(4), 424–433.

Armstrong, G.W. 2004. Sustainability of timber supply considering the risk of wildfire. *For. Sci.* **50**(5), 626–639.

Boychuk, D., and Martell, D.L. 1996. A multistage stochastic programming model for sustainable forest-level timber supply under risk of fire. *For. Sci.* **42**(1), 10–26.

CCFM 2003. *Defining Sustainable Forest Management in Canada. Criteria and Indicators. 2003.*. Canadian Council of Forest Ministers.

Gustafson, E.J., and Crow, T.R. 1998. Simulating spatial and temporal context of forest management using hypothetical landscapes. *Environmental Management* **22**, 777–787.

Johnson, E.A., and Gutsell, S.L. 1994. Fire frequency models, methods, and interpretations. In: *Advances in Ecological Research* (M. Begon and A. H. Fitter, Eds.). Vol. 25. pp. 239–287. Academic Press. London.

Johnson, K.N., and Scheurman, H.L. 1977. *Techniques for Prescribing Optimal Timber Harvest and Investment under Different Objectives – Discussion and Synthesis.* number 18 In: *Forest Science Monograph.* Society of American Foresters.

Martell, D.L. 1994. The impact of fire on timber supply in Ontario. *For. Chron.* **70**(2), 164–173.

MOSEK 2009. MOSEK optimization tools.

Parisien, M.A., Hirsch, K.G., Lavoie, S.G., Todd, J.B., and Kafka, V.G. 2004. Saskatchewan fire regime analysis. Inf. Rep. NOR-X-394. Northern Forestry Centre. Canadian Forest Service. Natural Resources Canada. Edmonton, AB, Canada. 49 p.

Reed, W.J., and Errico, D. 1986. Optimal harvest scheduling at the forest level in the presence of risk of fire. *Can. J. Forest Res.* **16**, 266–278.

Remsoft 2009. Woodstock.

The Montréal Process 1999. *Criteria and Indicators for the Conservation and Sustainable Management of Termperate and Boreal Forests*. 2nd ed.

Van Wagner, C.E. 1978. Age-class distribution and the forest fire cycle.. *Can. J. Forest Res.* **8**, 220–227.

Van Wagner, C.E. 1983. Simulating the effect of forest fire on long-term annual timber supply. *Can. J. Forest Res.* **13**, 451–457.

## Chapter 4

# A Case Study Application

## 4.1 Introduction

This chapter provides a case study application of the stylized model described in Chapter 3. Whereas Chapter 3 formally described a model and presented a simple hypothetical forest application, this paper serves as an extension by applying the model to actual forest inventory and stand yield data.

As in Chapter 3, this chapter will investigate the benefits a timber buffer stock will have on achieving various indicator targets (harvest level and old forest area). Further, the presence or absence of a buffer stock will be compared from a net present value perspective.

To avoid redundancy, the Chapter 3 model is not formally described here; the formal model is merely referenced where necessary. However, a description of the data used and the Woodstock (Remsoft, 2009) model formulation is provided.

## 4.2 Methodology

### Modeling System

The system of models used in this paper is very similar to that which was described in Chapter 3. The main structure still consists of an optimizing forest estate model used to choose optimal timber harvest plans, nested within a Monte Carlo model of stand replacing forest fires. However, with increasing complexity of input data the programming complexity of the modeling system increases. The full Python code for the modeling system used in this paper can be found in Appendix E.

The one difference to the Chapter 3 model is the addition of a mechanism to prevent infeasibility relating to the old forest area constraints. An infeasible solution is of no worth to the ultimate goal of plotting probability distributions for our modeled indicators. Therefore the Python script is required to revise the old forest constraints "on the fly" and attempt to reoptimize. In Chapter 2 a simple loop was created which incrementally lowered the harvest level constraint by 1% until a feasible solution could be found. This was an acceptable method when we know a single constraint is causing the infeasibility. With multiple forest types in an actual forest inventory there can be multiple old forest constraints. This approach then becomes very difficult to apply due to the fact that the script does not know which old forest constraint is causing the infeasibility. If all of the old forest constraints are lowered we would be moving further away from an optimal solution with each increment. Thus another approach was necessary for this case study.

Each time an infeasible solution is reached the python script reformats the problem using a goal programming approach. The goal programming approach is merely used to find the minimum feasible old forest constraints for each forest type present. The system then switches back to a conventional linear programming model for the final solution. If the old forest achieved in the goal programming model is lower than the original constraint, the LP model uses that achievement level as a constraint for that particular period. Otherwise the original constraint level is used. In order to ensure that the goal programming model achieves the highest possible old forest level the penalty wieghts were set very high (10,000,000). In the next planning period the original constraints are restored for all periods. The pseudo code for the modeling system is listed in Fig. 4.1.

### Modeled Indicators

As in Chapter 3, the modeled indicators that are examined in this application are harvest level, old forest area and net present value. One hundred draws from the modeling system produces a distribution of harvest and old forest area levels which are subsequently plotted and analyzed. In addition, the distribution of net present values for either strategy is presented and discussed.

### Monte Carlo burn proportion model

The fire regime model used for Chapters 2 and 3 is also used in this model. The model performs a random draw (with replacement) from a thirty-one year fire dataset (1977 to 2007) developed from the PA FMA area fire history. A complete description of the methodology used to create the area burned dataset is provided in Chapter 2.

### Inventory and Yields

The area of interest for this paper continues to be the PA FMA area in Saskatchewan, Canada. A recently completed planning inventory was obtained from the Saskatchewan Ministry of Environment's Forest Service Branch for use in this project. The planning inventory was a compilation of many spatial datasets overlaid for the purposes of identifying lands eligible for harvest within the license area.

The base dataset for creation of the planning inventory was the Saskatchewan Forest Vegetation Inventory (Saskatchewan Government, 2004). The average photo source year for the inventory was 2000. Updates for harvest and fire depletions were completed up to and including 2006/2007 and 2007/2008 respectively. Through the planning inventory

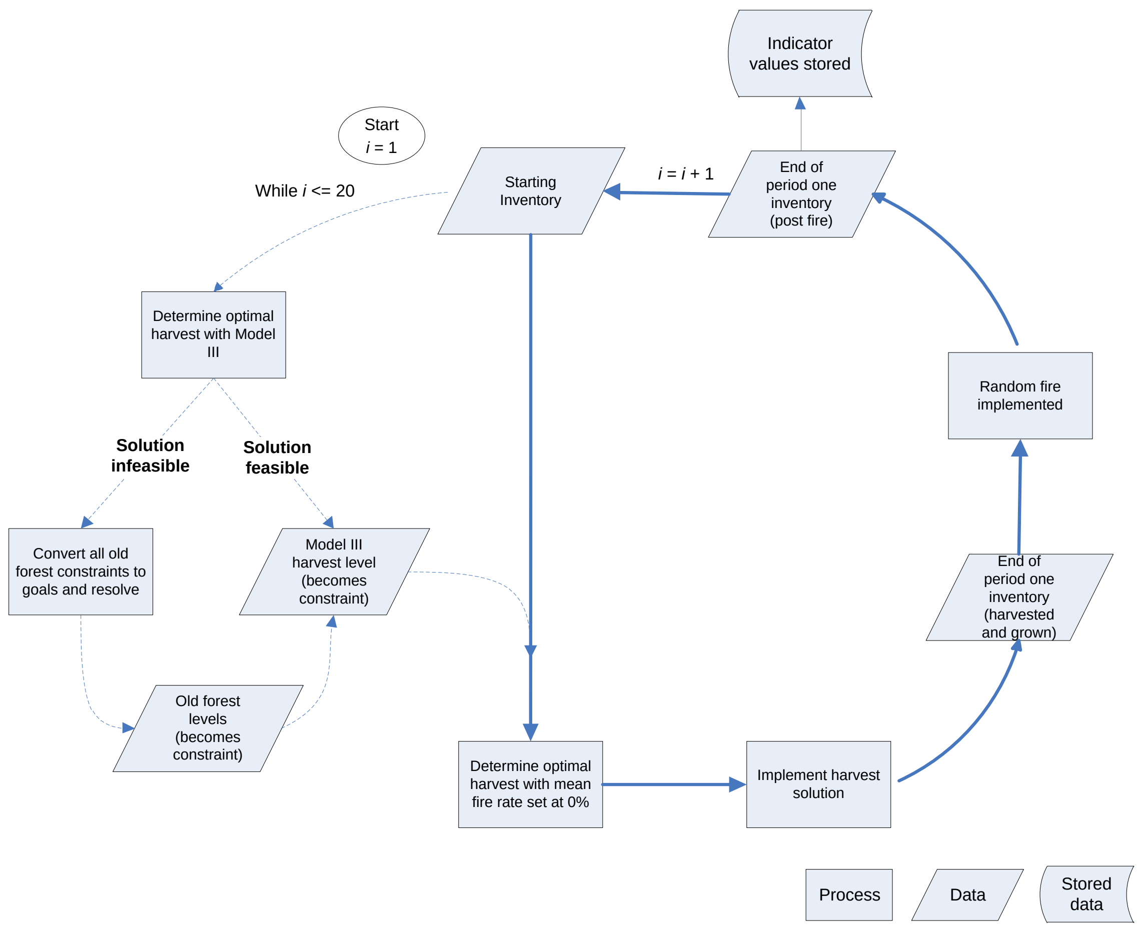



Figure 4.1: Overview of Chapter 4 Modelling System

development process forest area was assigned an administrative class depending on the spatial location. Table 4.1 provides an area breakdown of the administrative classes used. Only provincial forest was deemed eligible for harvest scheduling.

Table 4.1: PA FMA area netdown administrative classes

| Adminitstrative Areas | Area (hectares) |
|---:|---:|
| Indian Reserve | 2,242 |
| Park | 724 |
| Representative Area Network | 4,320 |
| Provincial Forest | 1,526,993 |
| Total | 1,534,279 |

Next, stand operability was identified. This included areas that fell within riparian buffers, inoperable slopes, non-productive and non-commercial forest. Table 4.2 provides a breakdown of the stand operability classes used. Although limited timber removal is allowed in the 30m riparian buffer class in the PA FMA area, our model allowed harvest scheduling in the net area class only.

Table 4.2: PA FMA area netdown stand operability classes

| Stand Operability Areas | Area (hectares) |
|---:|---:|
| Net area | 1,426,002 |
| 10m riparian buffer | 11,024 |
| 30m riparian buffer | 50,939 |
| Inoperable slope | 5,141 |
| Non-productive | 2,844 |
| Non-productive | 31,043 |
| Total | 1,526,993 |

Empirical natural stand yield curves were provided with the planning inventory (Timberline, 2008). Yield curves were developed for eighteen different yield groups, with yield group stratification based primarily on forest species composition. Further stratification by density class, site productivity class or forest management zone (roughly equivalent to ecoregion) was completed when supported by the data.

In total, 1,426,002 hectares of productive forest was eligible for harvest scheduling in the model. The areas by yield group are listed in Table 4.3.

The yield curve utilization specifications closely followed that of the last approved Forest Management Plan (FMP) for the PA FMA area (1999). The specifications are as follows:

- 30 cm stump height
- 8 cm inside bark top diameter for softwoods and hardwoods
- 5.0 m bole length (from stump height to point of top diameter limit)

Table 4.3: PA FMA area yield groups

| Yield Group Identifier | Yield Group Description | Area (hectares) |
|---|---|---|
| HW1 | Hardwoods (Site III) | 208,882 |
| HW2 | Hardwoods (Site I and II) | 166,031 |
| HJP | Hardwood leading mixedwood with jack pine | 51,907 |
| HXS1 | Hardwood dominated mixedwood with white spruce, balsam fir or black spruce (FMZ 1) | 30,351 |
| HXS2 | Hardwood dominated mixedwood with white spruce, balsam fir or black spruce (FMZ 2 and 3) | 63,061 |
| JPH | Jack pine leading mixedwood with hardwoods | 40,370 |
| XSH1 | White spruce, balsam fir, or black spruce dominated mixwood (FMZ 2 and 3) | 13,408 |
| XSH2 | White spruce, balsam fir, or black spruce dominated mixwood (FMZ 1) | 33,592 |
| BS1 | Black spruce dominated softwood (sites II and III) | 114,690 |
| BS2 | Black spruce dominated softwood (sites I) | 86,807 |
| JP1 | Jack pine dominated softwood (site III) | 53,068 |
| JP2 | Jack pine dominated softwood (sites I and II) | 254,314 |
| JPBS1 | Jack pine and black spruce dominated softwood (crown closure B) | 28,731 |
| JPBS2 | jack pine and black spruce dominated softwood (crown closure C and D) | 171,936 |
| WSBF1 | White spruce or balsam fir dominated softwood (FMZ 2 and 3) | 14,856 |
| WSBF2 | White spruce or balsam fir dominated softwood (FMZ 1) | 36,948 |
| TL1 | Stands with at least 11% tL species composition (FMZ 1) | 14,430 |
| TL2 | Stands with at least 11% tL species composition (FMZ 2 and 3) | 42,621 |
| Total | | 1,426,002 |

The age class distribution of the PA FMA area planning inventory could be described as irregular which would be expected given the stochastic nature of the fire regime. The age class distribution does not resemble the hypothetical forests we have examined earlier nor does it resemble a distribution that would result from a negative exponential survivorship curve (Van Wagner, 1978; Armstrong, 1999). Fig. 4.2 illustrates the current age class distribution of the PA FMA area. Upon examination of the age class graph it is evident there is a large amount of recently disturbed area as well as older forest classes. The age classes in between range from abundant to almost entirely absent. Note the relatively small amount of area between age classes 40 to 60.

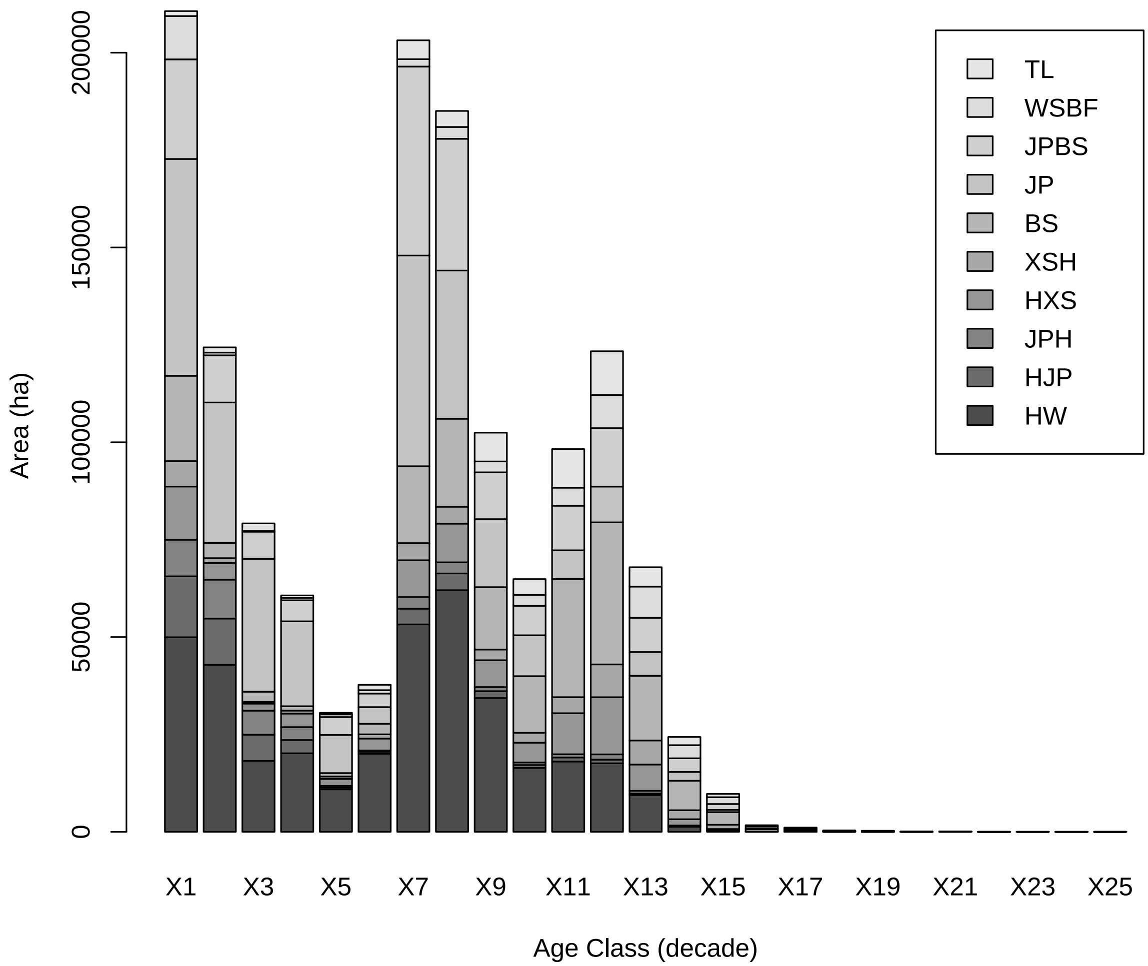



Figure 4.2: PA FMA area age class distribution

### Woodstock Model Formulation

A complete Woodstock model for the PA FMA area was also provided by the Forest Service Branch for use in the project. The model was created with the intent of approximating the management assumptions used in the last 20 year PA FMA area forest management plan (FMP) approved in 1999. This model served as a starting point and was adapted accordingly to be used in our analysis.

The final Woodstock model used in this exercise became a hybrid of the model described in Chapter 3 and the PA FMA model provided from the Forest Service Branch. Relative to the Chapter 3 model there was an increase in complexity to account for 18 different yield groups and yield group specific renewal cost data.

The model description that follows assumes the reader is familiar with Woodstock terminology. The complete model formulation in Woodstock syntax is presented in Appendix F.

#### Objective Function

The model objective function maximized present value revenue minus present value cost over the 200 year planning horizon modeled. A discount rate of 5% was applied with the assumption that all activities occured at the mid point of each ten year planning period.

#### Even Flow Constraints

Separate even flow constraints were applied to softwood and hardwood volume to ensure volume flow stability throughout the planning horizon. This differed from the Forest Service Branch model which allowed considerable variation in volume flows. For the sake of simplicity it was felt the Chapter 3 even-flow approach was more suitable. Future sensitivity analyses should include incorporating the Forest Service Branch flow constraints.

#### Non-declining Growing Stock Constraint

A non-declining operable growing stock constraint was applied to the last 50 years of the planning horizon to ensure the model did not liquidate growing stock. This was not present in the Forest Service Branch model. The Forest Service Branch model included a constraint which forced the model to maintain at least 14.4 years of operable growing stock throughout the planning horizon. This constraint obviously conflicted with the intent of our analysis and was therefore dropped.

#### Old Forest Constraints

With the Chapter 3 model only one old forest constraint was necessary due to the use of a hypothetical single species forest. The forest modeled in this application is composed of many forest types as reflected by the multiple yield groups previously identified. In order to evenly maintain old forest with the presence of multiple forest types, a series of constraints were necessary. This amounted to ten separate old forest constraints, one for each forest type present. For example, the BS1 and BS2 yield groups had one combined old forest constraint for the BS forest type.

In the original 1999 FMP separate old and very old seral stage targets were used. The Forest Service Branch model had two sets of constraints to reflect this. For this analysis we use a single set of "old forest" constraints which will account for both of the old and very old seral stage constraints.

Our old forest constraint levels equated to the combined old and very old seral stage targets used in the 1999 FMP. With the exception of the white spruce leading forest type (WSBF), all strata had old forest constraints of 6%. The white spruce leading forest type had a target of 12%. The old forest constraints used in our analysis are provided in Table 4.4.

Table 4.4: Old forest targets

| Forest Types | % Old Forest Target |
|---:|---:|
| HW | 6 |
| HJP | 6 |
| HXS | 6 |
| JPH | 6 |
| XSH | 6 |
| BS | 6 |
| JP | 6 |
| JPBS | 6 |
| WSBF | 12 |
| TL | 6 |

Due to the differing growth and development patterns of forest types the age classes that define old forest varied. Table 4.5 provides the definitions used for old forest.

Table 4.5: Age definitions for old forest

| Forest Type | Old Forest (Years) |
|---:|---:|
| HW | 80+ |
| HJP | 80+ |
| HXS | 80+ |
| JPH | 80+ |
| XSH | 100+ |
| BS | 110+ |
| JP | 80+ |
| JPBS | 110+ |
| WSBF | 100+ |
| TL | 110+ |

### Area Burned Constraint

A model III formulated using Woodstock requires a constraint forcing the model to undergo a burn action/transition combination. This action/transition combination (described in greater detail further on) resets the age of a proportion of each age class in every period. The proportion is equal to the burn fraction. The remaining proportion (1-burn fraction) retains the original age. The constraint associated with this burn action/transition ensures that all area in the harvestable inventory will be subject to this burn action/transition combination in every period.

All other model III starting inventory and area transfer constraints were applied as described in Chapter 3 and will not be described again here.

### Actions and Transitions

The model contains three possible "actions": harvest, death and burn. The harvest action logically equates to the harvest of forest area. Stands were deemed operable for harvest if they held a volume $\geq$ 60 $m^3$/ha. Although the Forest Service Branch model also had minimum age operability limits, our model did not. This was due to the differing objective functions used. The Forest Service Branch model maximized total harvest volume while our model objective function had a financial basis. With our objective function the minimum age of harvest would be determined by the yields and associated financial parameters. The transition for a harvest action in our model is to revert back to the pre-disturbance development type at age zero.

The death action models the natural break-up or senescence of a stand. Table 4.6 lists the break-up ages used for each yield group in the model. The transition for a death action is to revert back to the pre-disturbance development type at age zero. Natural break-up was not included in the theoretical models presented in Chapters 2 or 3. However, it was included in the Forest Service Branch model and was therefore deemed a necessary component of this case study.

Table 4.6: Death ages for forest types

| Forest Type | Death Age (Years) |
|---|---|
| HW | 130 |
| HJP | 130 |
| HXS | 140 |
| JPH | 140 |
| XSH | 150 |
| BS | 180 |
| JP | 150 |
| JPBS | 160 |
| WSBF | 170 |
| TL | 150 |

Finally, the burn action is our mechanism to incorporate a constant periodic burn rate into the model. This is a feature of a model III formulation. Area of all age classes are available to undergo the burn action. Once area undergoes the burn action the associated transition reverts y% of the area to the pre-disturbance development type at age zero and 100-y% of the area retains its original age. In association with the area burnt constraint previously described, this action-transition combination forces the model to reset the ages of an amount of area equal to the burn fraction every period.

**Financial Parameters**

A rate of 5% was chosen for discounting model costs and revenues. The revenue from timber sales remained $50/m^3$. The costs were separated into a harvest cost and a renewal cost. The harvest cost remained $5000/ha. The renewal costs were updated to reflect the yield group harvested. General estimates for renewal costs for each forest type were provided by the Forest Service Branch (David Stevenson, Saskatchewan Forest Service, personal communication, December 29, 2009) and are illustrated in Table 4.7.

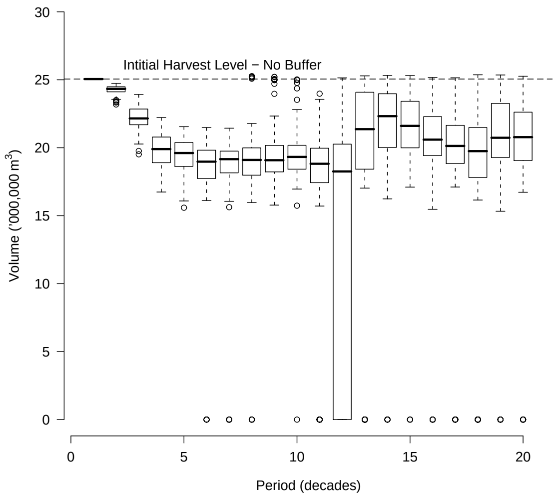
Table 4.7: Estimated renewal costs for forest types

| Forest Type | Estimated Renewal Cost ($/ha) |
|---|---|
| HW | 18.00 |
| HJP | 88.00 |
| HXS | 397.00 |
| JPH | 158.00 |
| XSH | 775.00 |
| BS | 775.00 |
| JP | 156.00 |
| JPBS | 311.00 |
| WSBF | 775.00 |
| TL | 775.00 |

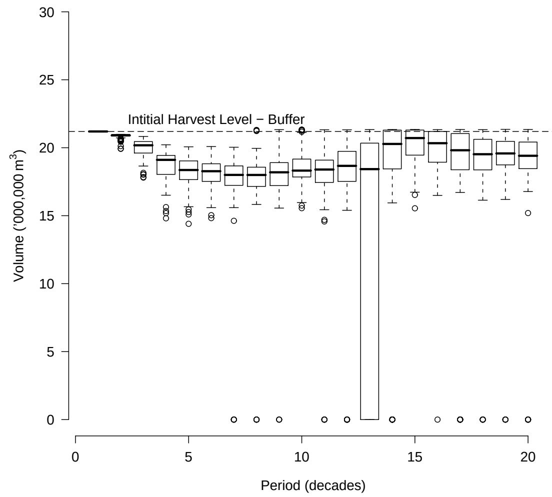
## 4.3 Results

The probability distributions for each of the indicators we have tracked are provided in Fig. 4.3 through to Fig. 4.14. The median, upper and lower quartiles are represented by the box. The whiskers represent the largest or smallest non-outlier value and the circles represent outliers.

The harvest level results are presented first in Fig. 4.3. Fig. 4.3(a) and Fig. 4.3(b) represent the distribution of harvest levels without and with a harvest buffer, respectively. The dashed line represents the initial even-flow harvest level that results from solving the optimization model outside of the modeling system.



(a) No Buffer Harvest Levels



(b) Buffered Harvest Levels

Figure 4.3: Harvest results for the no buffer and buffer strategies

Fig. 4.4 through to Fig. 4.13 present the old forest level distributions for each of the forest types modeled. The dashed line represents the old forest target which corresponds with Table 4.4.

Subfigures Fig. 4.14(a) and Fig. 4.14(b) represent the cumulative 200-year harvest and net present value distributions, respectively.

## 4.4 Discussion

Intrepretion of the results tend to be somewhat more challenging with the increased data complexity, although some trends appear evident when the data and model formulation are considered. Starting with the harvest levels, some similarities with Chapter 3 can be seen. Obviously the non-buffered harvest level is considerably higher at the outset of the planning horizon. With a cursory glance, it appears that harvest levels continue to be higher or at least comparable throughout the planning horizon without a buffer. These results are similar to the "surplus" forest used in Chapter 3. This similarity may be expected given that the starting age class distribution of the PA FMA area possesses a large surplus of mature timber.

Of interesting note and not present in chapter 3 is the possibility of a catastrophic drop in harvest levels in period 12 for the no buffer scenario and period 13 for the buffered scenario. Looking back at the starting age class distribution provides some insight into this result. The significant gap in the age class distribution from ages 40 to 60 is likely contributing to this drop.

A key difference between our case study model and Chapter 3 was the addition of stand

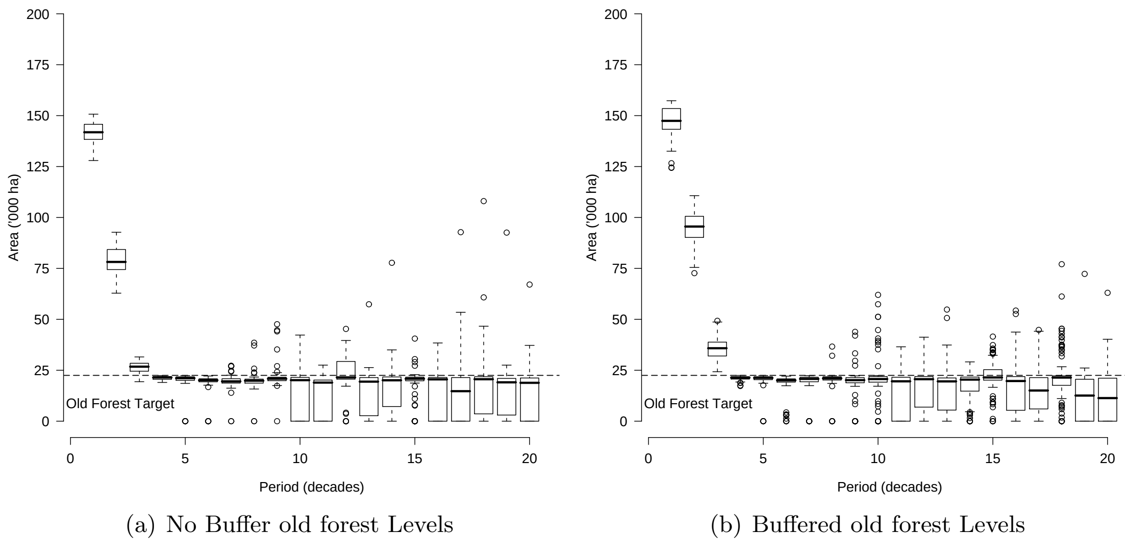



Figure 4.4: HW old forest results for the no buffer and buffer strategies

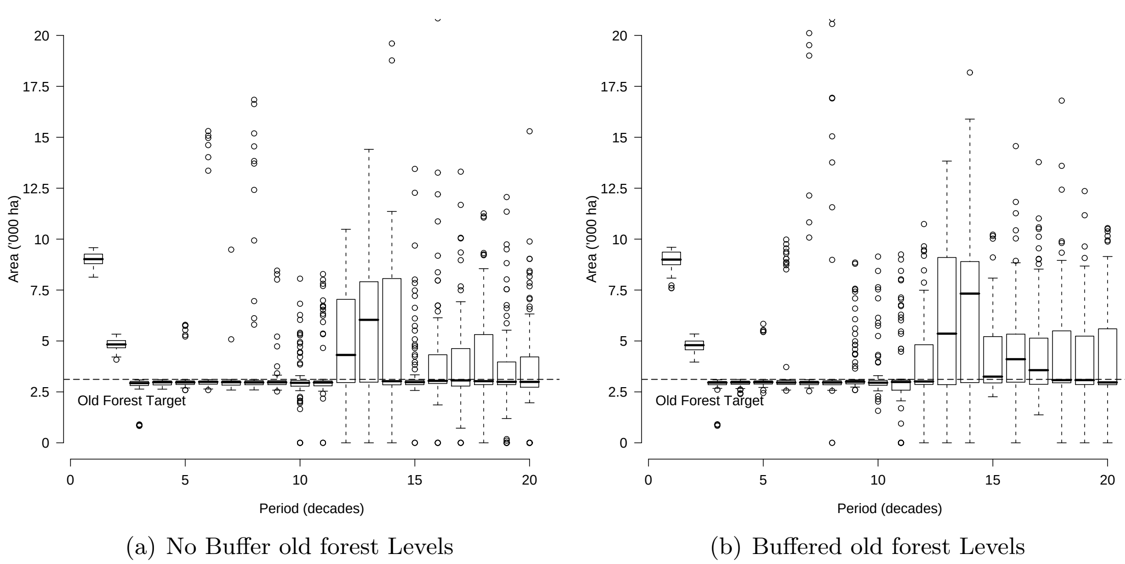
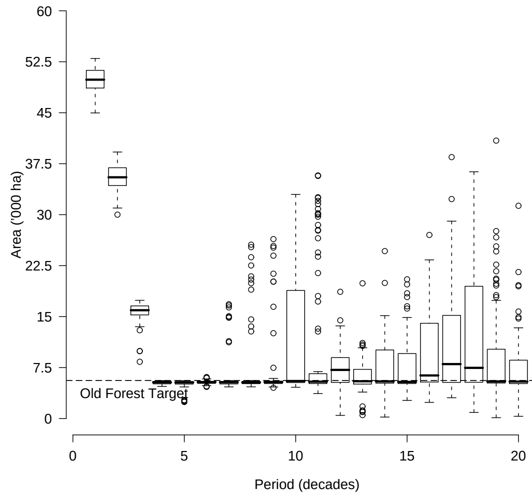
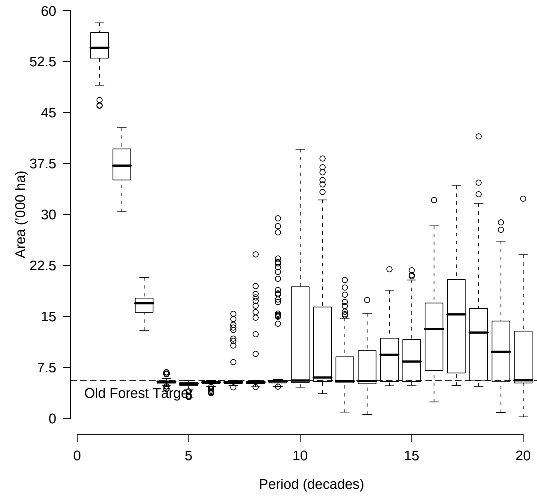



Figure 4.5: HJP old forest results for the no buffer and buffer strategies

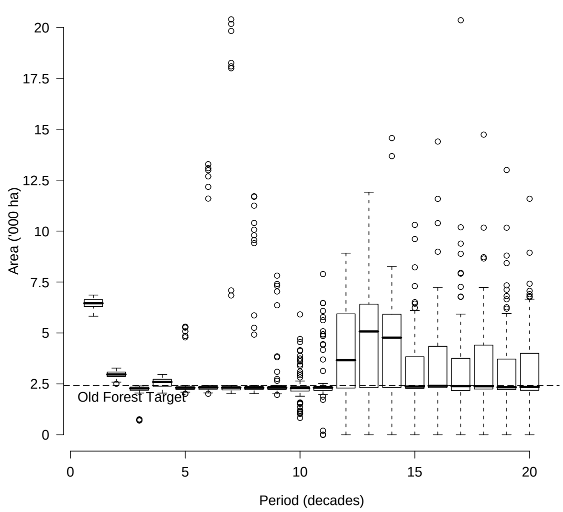



(a) No Buffer old forest Levels

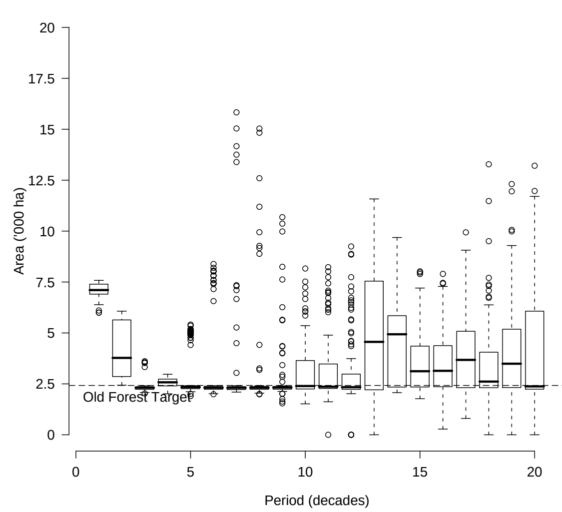



(b) Buffered old forest Levels

Figure 4.6: HXS old forest results for the no buffer and buffer strategies

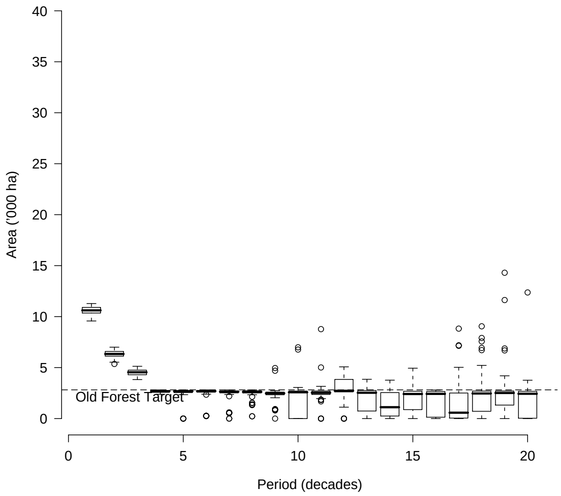



(a) No Buffer old forest Levels

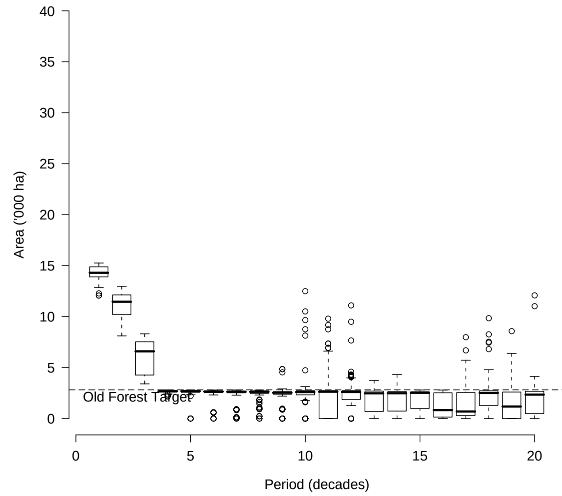



(b) Buffered old forest Levels

Figure 4.7: JPH old forest results for the no buffer and buffer strategies

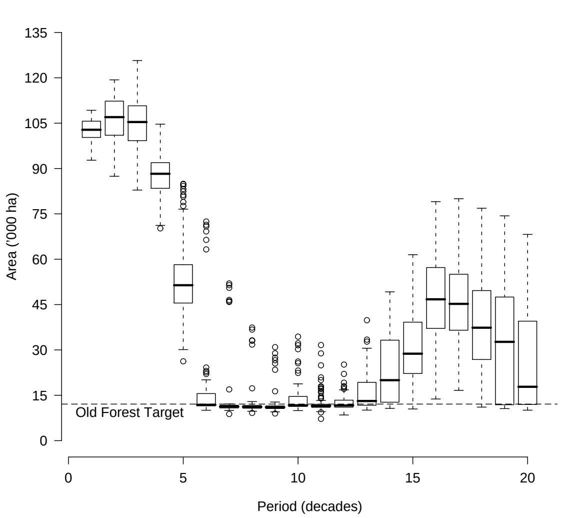



(a) No Buffer old forest Levels

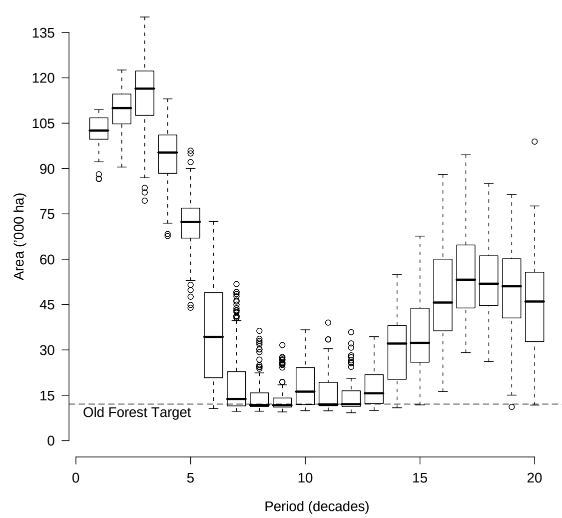



(b) Buffered old forest Levels

Figure 4.8: XSH old forest results for the no buffer and buffer strategies

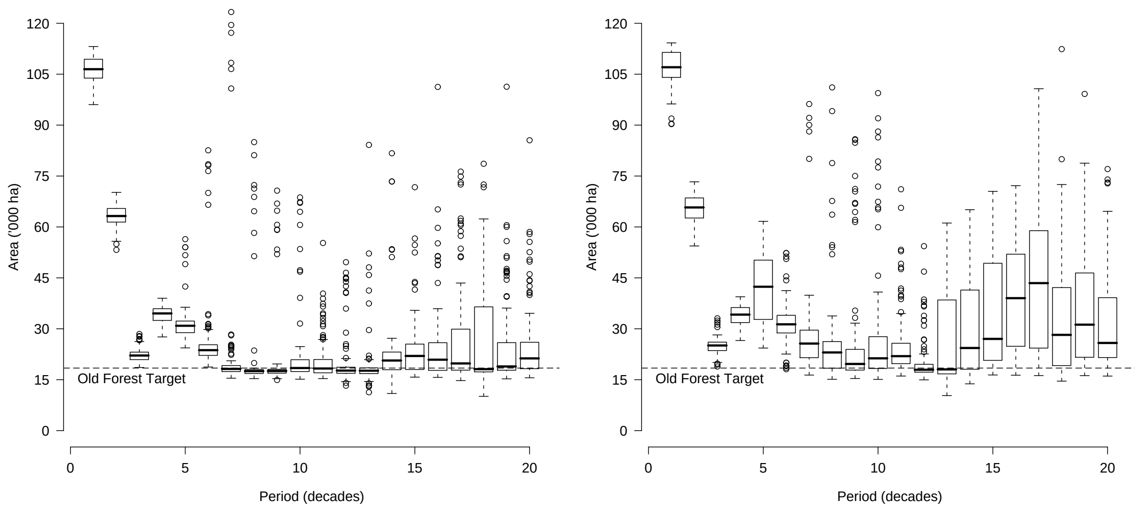



(a) No Buffer old forest Levels



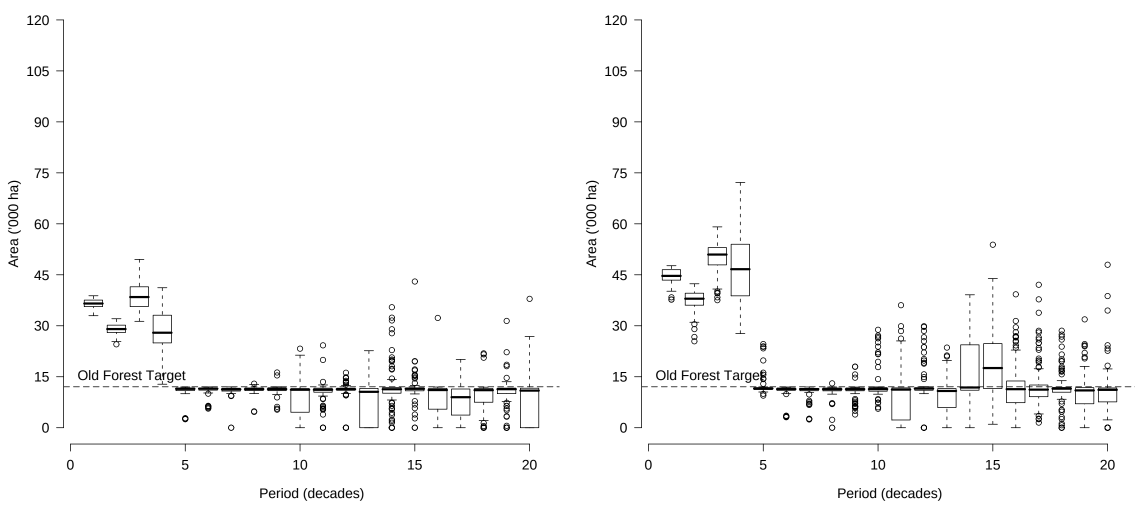

(b) Buffered old forest Levels

Figure 4.9: BS old forest results for the no buffer and buffer strategies



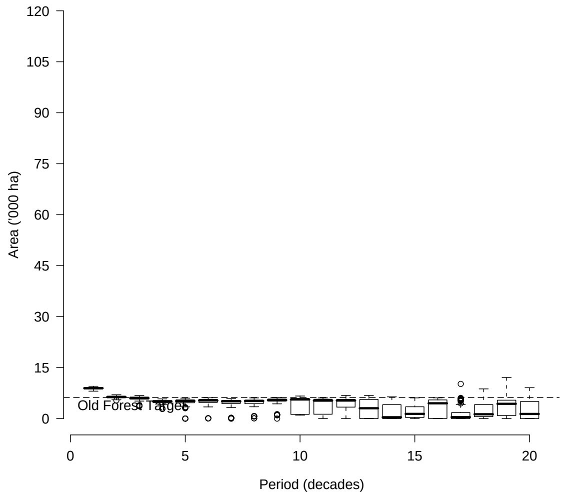

(a) No Buffer old forest Levels (b) Buffered old forest Levels

Figure 4.10: JP old forest results for the no buffer and buffer strategies

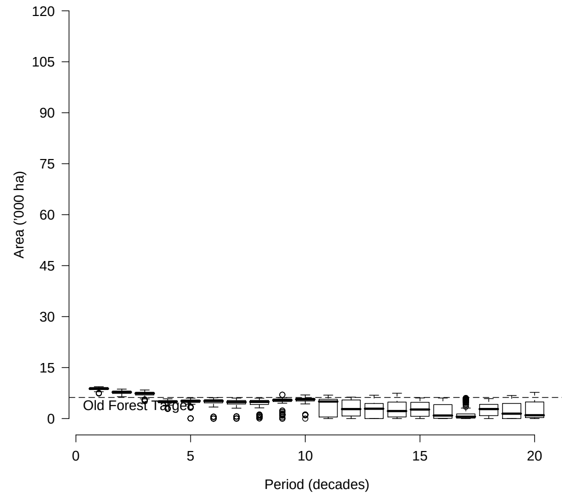



(a) No Buffer old forest Levels (b) Buffered old forest Levels

Figure 4.11: JPBS old forest results for the no buffer and buffer strategies

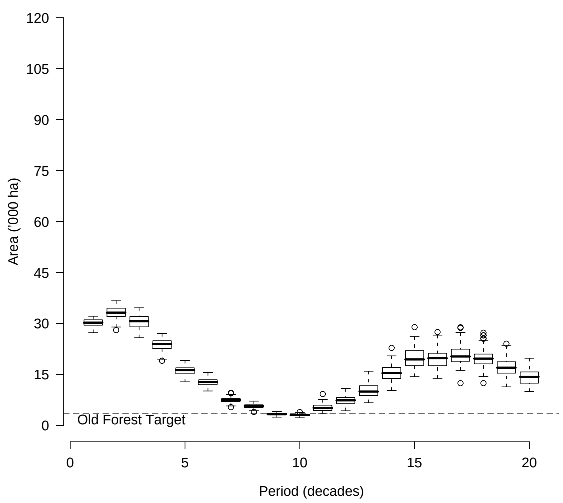



(a) No Buffer old forest Levels

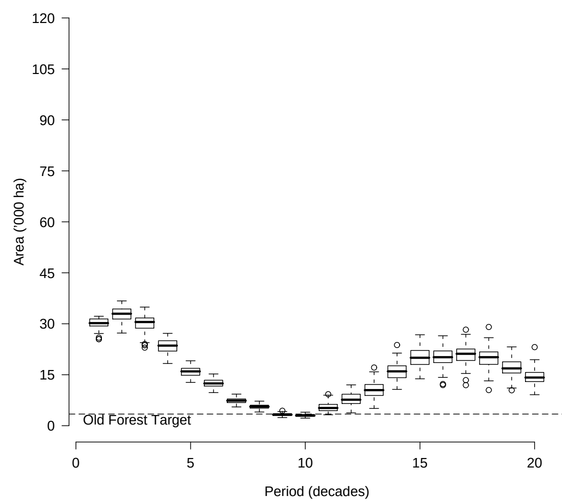



(b) Buffered old forest Levels

Figure 4.12: WSBF old forest results for the no buffer and buffer strategies

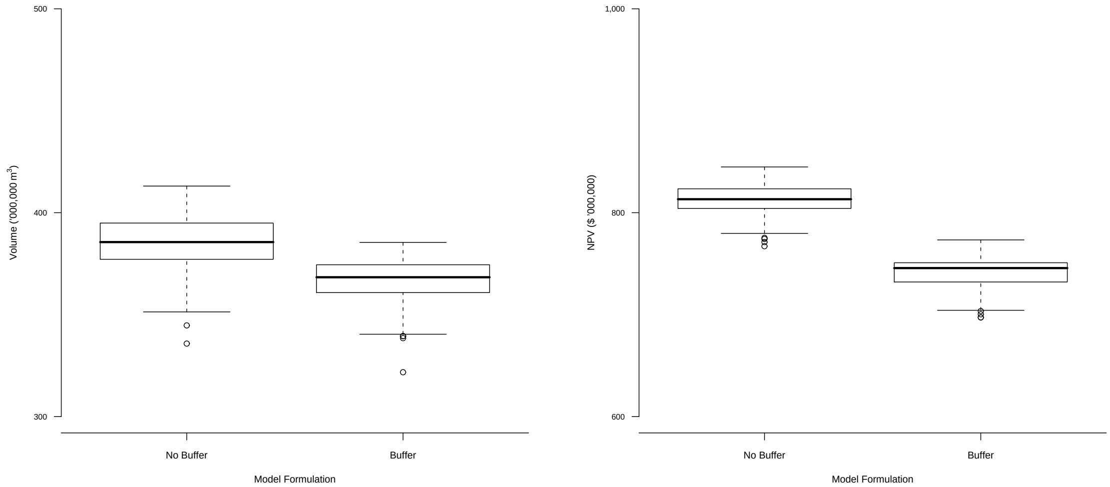



(a) No Buffer old forest Levels




(b) Buffered old forest Levels

Figure 4.13: TL old forest results for the no buffer and buffer strategies




(a) Cumulative Harvest Levels

(b) Net Present Value

Figure 4.14: Cumulative harvest and NPV for the no buffer and buffer strategies

break-up ages. As discussed more further on, the use of stand break up ages may be a major driver in the results of this exercise. With respect to the age class gap, the stand break up ages are likely causing considerable stand mortality just prior to this point in the planning horizon. So much so that under a severe combination of fire years it is no longer profitable to harvest timber in this forest 120 to 130 years into the future using an even-flow and maximize net present value model formulation. The forest is simply too young to harvest, likely resembling a deficit forest. A deficit forest was intentionally left out of the Chapter 3 analysis for this same reason.

It is challenging to tease out any trends when comparing the old forest distributions for the two strategies. Although some slight differences are evident, generally the two sets of distributions appear similar. This result was surprising considering Chapter 3 pointed towards a noticeable improvement in old forest levels with the presence of a buffer stock. Once again the stand break up ages may have contributed significantly to this result. With a relatively large proportion of the starting age class distribution being mature, a constrained harvest level does not necessarily provide an increase in old forest. In fact, it may result in higher levels of mortality in the model which tempers the increase of old forest area. There is very little opportunity to harvest this area prior to break up due to the constrained harvest level.

A sensitivity analysis was completed by running the model without stand break-up ages. The results indicated that old forest areas tended to be higher when a buffer was present. This would support the notion that incorporting a buffer while modelling stand break-up ages may result in greater mortality and not necessarily a significant increase in old forest area.

The validity of using stand break-up ages may come in to question with this result. Without a doubt, forest dynamics are much more complicated then the use of stand break ages would suggest (Cumming *et al.* (2000)). It should be noted, however, that without accounting for stand decline in a model an unrealistic amount of old forest may be reported for some stand types. It is debatable which approach provides a better representation of actual stand dynamics.

Given the harvest results of the two strategies it comes as no surprise that the cumulative 200 year harvest and NPV levels are generally higher for the no buffer strategy. If one were to use NPV as a decision criteria the no buffer strategy is the clear winner at a 5% discount rate. The greater harvest levels early in the planning horizon would contribute to this result. Using NPV as a decision criterion favours near-term over long-term return.

In Chapter 3 a hypothetical single species forest was used to demonstrate the assessment of a buffer stock strategy. The single species forest allowed a relatively straightforward comparison of the alternative strategies tested. In reality a single species forest is unlikely in the boreal forest. Thus is was felt a case study application was necessary. For the sake of consistency the PA FMA area was considered suitable for a case study. Not surprisingly, the forest in the PA FMA area proved to be much more complicated than a single species forest with a simple age class distribution. The PA FMA area is composed of multiple forest types with different development patterns as reflected in the 18 different yield groups. The age class distribution of these forest types also proved to be much more complex than previous examples used. In order to account for this increase in complexity our optimization model required an increased number of constraints. Our modeling system (Python script) also required increased complexity. Overall though, the increased complexity proved to be very much manageable. The principles behind the modeling system remained identical to Chapter 3 and the approach appears as though it would be worthwhile in a real world management context.

# Bibliography


Armstrong, G.W. 1999. A stochastic characterisation of the natural disturbance regime of the boreal mixedwood forest with implications for sustainable forest management. *Can. J. Forest Res.* **29**(4), 424–433. doi:10.1139/cjfr-29-4-424.

Cumming, S.G., Schmiegelow, F.K.A., and Burton, P.J. 2000. Gap dynamics in boreal aspen stands: is the forest old than we think?. *Ecol. Appl.* **10**(3), 744–759.

Remsoft 2009. Woodstock.

Saskatchewan Government 2004. Saskatchewan forest vegetation inventory. 83p.

Timberline. 2008. Developing yield curves for the Prince Albert Forest Management Agreement area. Technical report. Timberline Natural Resource Group. Prince Albert, SK, Canada. 99 p.

Van Wagner, C.E. 1978. Age-class distribution and the forest fire cycle.. *Can. J. Forest Res.* **8**, 220–227.

# Chapter 5

# Conclusion

This thesis presented three papers that investigated the impacts stochastic wildfire may have on mulitple values in the boreal mxedwood forest of Saskatchewan, Canada. A probabilistic sustainability approach was taken which builds on the work described in Armstrong (2004).

The first paper (Chapter 2) focused on the impact a reduced harvest level would have on the probability of achieving harvest and old forest targets. In the modelling system a maximin old forest objective function was used subject to a minimum harvest level constraint. The harvest level constraint was set using proportions of a baseline harvest level. The baseline harvest level was calculated using a simple timber supply model which maximized harvest level subject to a minimum old forest constraint in the abscence of fire disturbance. The results suggested that a reduced harvest level will increase the probability of meeting harvest and old forest targets for the hypothetical single species forests examined. However, a very significant harvest level decrease was necessary.

The second paper (Chapter 3) also examined the impact of a reduced harvest level. However, the approach taken in this paper included an economic element and a model formulation that accounted for the mean annual disturbance rate. The economic element was introduced through the use of a maximize net present value objective function. The mean annual fire disturbance rate was directly incorporated into the optimization model (sometimes referred to as a model III formulation). Overall, the modeling system followed a pattern which simulated the cycle of determining a harvest schedule which anticipated the impact of the mean disturbance rate, implementing the harvest schedule and finally burning the forest according to the stochastic disurbance regime. This resulted in the allowance for a stock of timber which would theoretically buffer the impact of fire disturbance. The probability distribution of net present values, cumulative 200 year harvest levels, periodic harvest and old forest area achievement were compared for scenarios with and without the buffer stock. Similar to Chapter 2 it was determined that a reduced harvest level could increase the probability of harvest and old forest target achievement given the hypothetical forest and model formulation used. Once again, the reduction would need to be significant to be effective. Overall, the decision whether the cost (immediate harvest level reduction) is worth the benefit (higher probability of target achievement) would need to be made by the forest stakeholders. From a net present value perspective it is likely that a no buffer stock strategy would be more profitable for five of the six discount rate / forest combinations examined. Cumulative 200 year harvest level is likely to be higher without a buffer stock on all six of six discount rate / forest combinations examined.

The third paper (Chapter 4) built on the work completed in Chapter 3 by applying the approach to actual forest inventory and yield data. A completed net down planning inventory, yield tables and a Woodstock model were obtained from the Saskatchewan Ministry of Environment's Forest Service Branch for use in the project. A modeling system very similar to that which was used in Chapter 3 was built using this information. The results of the analysis were somewhat surprising. Although harvest was generally higher without the presence of a buffer, old forest achievement was comparable between strategies. A key driver in the results appeared to be the use of stand break-up ages which were not used in the previous chapters. With the inclusion of stand break-up ages a reduced harvest level did not improve indicator achievement, it merely resulted in higher levels of stand mortality.

The approaches described here may act as an avenue for assessing risk and could allow stakeholders the ability to recognize future implications for their decisions. The risk assessment process may be composed of two main components: the likely values of projected indicators and the consequence of that particular indicator value. This analysis approach could provide the first component, the likely values of projected indicators (given a set of modeled management assumptions). Another body, perhaps composed of stakeholders, could determine the consequences of the projected indicator values.

# Bibliography

Armstrong, G.W. 2004. Sustainability of timber supply considering the risk of wildfire. *For. Sci.* **50**(5), 626–639.

# Appendices

# Appendix A

# Chapter 2 Modeling System in Python Code

```
# Chapter 2 modelling system
# Author:  Matt Conrod
# Date: February 20th, 2010

import os

# opening and reading random fire file
import string
population=open("C:\\MC_CHP1_model\\Area_burned_PAFMA_Jan_08.txt","r")
fire_list=map(string.strip, population.readlines())
population.close()

# Defining a "sample with replacement" function to select random fire years
# credit author(s) of random.py
import random

def sample_wr(population, k):
    "Chooses k random elements (with replacement) from a population"
    n = len(population)
    _random, _int = random.random, int  # speed hack
    result = [None] * k
    for i in xrange(k):
        j = _int(_random() * n)
        result[i] = population[j]
    return result

# The draw loop
draw = 1
while draw <=100:

    #refreshing the .are file
    os.system("C:\\MC_CHP1_model\\Refresh_Area.bat")

    print "Draw # "
    print draw

    #The planning period loop
    planning_period = 1
    while planning_period <=20:

        proportion=1.00
        Harv_constraint=24062151#input harvest request here
        Opt_str_flt_edit=24062151#input harvest request here

        #refreshing the .opt file
        os.system("C:\\MC_CHP1_model\\refresh_Opt.bat")
```

```
loop = 1
while loop <= 1:

    exists=0

    print "Planning Period "
    print planning_period

    #Generating matrix, solving matrix and generating reports
    os.system("C:\\MC_CHP1_model\\call_Opt_OG_1.bat")

    #Opening and reading end of period one harvest total report
    #from Woodstock
    f=open("c:\\MC_CHP1_model\\HARVREPORT.TXT","r")
    harv_tot=map(string.strip, f.readlines())
    f.close()

    #deleting text line from start of HARVREPORT.TXT file data
    del harv_tot[0:4]
    harv_totsplt_temp= [i.split() for i in harv_tot]
    del harv_totsplt_temp[1]

    #Deleting first variables in HARVREPORT.TXT file so that only harvest
    #volume is remaining
    harv_totsplt=harv_totsplt_temp[0]
    del harv_totsplt[0]
    del harv_totsplt[0]
    del harv_totsplt[0]
    harv_totsplt_str=str(harv_totsplt[0])
    harv_totsplt_str_2=harv_totsplt_str.replace(",","")
    harv_totsplt_flt=eval(harv_totsplt_str_2)
    print 'Current harvest accomlishment: ', harv_totsplt_flt

    if os.path.exists("c:\\MC_CHP1_model\\Infeasible.msg"):
        exists = 1

    #if infeasible - identified by infeasible msg or the harvest acheivement
    #being lower than the request
    if exists == 1 or harv_totsplt_flt < Opt_str_flt_edit:
        proportion=proportion-.01
        print proportion, '%'

       # Opening and reading OG_1.opt from Woodstock
       #and stripping file down to harvest request (constraint) value only
        f=open("C:\\MC_CHP1_model\\OG_1.opt","r")
        Opt_sect=map(string.strip, f.readlines())
        f.close()

        del Opt_sect[0:5]
        del Opt_sect[1:5]

        # Spliting string data into lists
        import re
        Opt_sect_splt = [i.split() for i in Opt_sect]
        Opt = Opt_sect_splt[0]
        del Opt[0:2]
        del Opt[1]
        Opt_str=str(Opt[0]) #only the harvest request figure remaining
                #from .opt file
        Opt_str_2=Opt_str.replace(",","")
        Opt_str_flt=eval(Opt_str_2)
        print 'Harvest request: ', Opt_str_flt

        #Calculating the new harvest request value with the
        #"Harv_constraint" value mulitplied by the "proportion" value
        Opt_str_flt_edit=Harv_constraint*proportion
        print 'New harvest request: ', Opt_str_flt_edit
```

```
        #rebuilding the .opt file with new harvest request
        f=open("c:\MC_CHP1_model\\OG_1.opt","w")
        f.write('; Optimize\n *OBJECTIVE\n _MAXMIN oldgrowtharea \
                1.._LENGTH\n \n *CONSTRAINTS\n Totvol >= ')
        new_harv=str(Opt_str_flt_edit)
        f.write(new_harv)
        f.write(' 1.._LENGTH\n \n _EVEN(totvol) 1.._LENGTH\n _NDY(standinginv) \
                 15.._LENGTH\n \n *FORMAT MOSEK')
        f.close()

        loop = loop + 0

    if exists == 0:
        loop = loop + 1

# generating 10 random fire years and summing to represent total
#percent area burned for the 10-year planning period
periodic_fire = sample_wr((fire_list),10)
print "Randomly drawing annual fire proportions....."
periodic_fire[0]=eval(periodic_fire[0])
periodic_fire[1]=eval(periodic_fire[1])
periodic_fire[2]=eval(periodic_fire[2])
periodic_fire[3]=eval(periodic_fire[3])
periodic_fire[4]=eval(periodic_fire[4])
periodic_fire[5]=eval(periodic_fire[5])
periodic_fire[6]=eval(periodic_fire[6])
periodic_fire[7]=eval(periodic_fire[7])
periodic_fire[8]=eval(periodic_fire[8])
periodic_fire[9]=eval(periodic_fire[9])

total_fire =1-((1-periodic_fire[0])*(1-periodic_fire[1])*(1-periodic_fire[2])
                  *(1-periodic_fire[3])*(1-periodic_fire[4])*(1-periodic_fire[5])
                  *(1-periodic_fire[6])*(1-periodic_fire[7])
                  *(1-periodic_fire[8])*(1-periodic_fire[9]))
print "Summing annual fire proportions for planning period...."
print total_fire

# Opening and reading end of period one age-class report from Woodstock run
f=open("C:\\MC_CHP1_model\\ageclassp1.txt","r")
Prd1_AgeC=map(string.strip, f.readlines())
f.close()

# deleting text lines from start of file data
del Prd1_AgeC[0:5]

# Spliting string data into lists
import re
Prd1_AgeC_splt = [i.split() for i in Prd1_AgeC]

# Creating a dictionary (array) indexed by planning period
dict_Prd1_AgeC=dict(Prd1_AgeC_splt)

# Creating a dictionary with area converted to float
dict_Prd1_AgeC_2 = dict(zip(dict_Prd1_AgeC.keys(), ([float(value) for value \
                in dict_Prd1_AgeC.values()])))
#print dict_Prd1_AgeC_2

# Convert keys to integer
dict_Prd1_AgeC_3 = dict(zip(([int(key) for key in dict_Prd1_AgeC_2.keys()]), \
               dict_Prd1_AgeC_2.values()))
#print dict_Prd1_AgeC_3

# calculating burn area
Prd_burn_areas = dict(zip(dict_Prd1_AgeC_3.keys(), \
               ([total_fire*value for value in dict_Prd1_AgeC_3.values()])))
print "Calculating burn areas...."
#print Prd_burn_areas
```

```
        # summing burn area
        total_burn_area = sum(Prd_burn_areas.values())
        print "Summing burn areas..."
        print total_burn_area

        #calculating new areas
        post_burn_areas = dict(zip(dict_Prd1_AgeC_3.keys(), \
([(value_1-(total_fire*value_1)) for value_1 in
                                        dict_Prd1_AgeC_3.values()])))
        print "Calculating new post-burn areas..."
        #print post_burn_areas

        # Adding burn areas to age-class one
        post_burn_areas[1] = post_burn_areas[1]+ total_burn_area
        print "Adding burn areas to age-class one...."
        #print post_burn_areas

        #Generating new .are file post-burn
        f=open("c:\\MC_CHP1_model\\OG_1.are","w")
        for key in sorted(post_burn_areas):
            f.write('*A pine ''%s %s\n' % (str(key), \
''.join(repr(post_burn_areas[key]).strip('[]').split(','))))
        f.close()

        #Deleting area younger then 90 yrs
        AGE_DEL = 1
        while AGE_DEL <=9:
            if post_burn_areas.has_key (AGE_DEL):
                del post_burn_areas[AGE_DEL]
            AGE_DEL=AGE_DEL+1

        #Calculating old-growth area after burn
        og_area=sum(post_burn_areas.values())
        print "calculating old-growth areas...."
        print og_area

        #writing the results file - [draw_#] [planning_period]
        #[OG_area] [harvest volume]
        og=open(r'c:\MC_CHP1_model\OG_results.txt','a')
        str_pp=str(planning_period)
        str_draw=str(draw)
        str_og_area=str(og_area)
        str_harv_totsplt=str(harv_totsplt[0])
        str_total_fire=str(total_fire)
        og.write(str_draw)
        og.write(" ")
        og.write(str_pp)
        og.write(" ")
        og.write(str_og_area)
        og.write(" ")
        og.write(str_harv_totsplt)
        og.write(" ")
        og.write(str_total_fire)
        og.write("\n")
        og.close()

        planning_period = planning_period + 1

    draw = draw+1
```

# Appendix B

# Chapter 2 Woodstock Model Formulation

```
ACTIONS

*ACTION harvest Y Harvest
  *OPERABLE harvest
    ? _CP >= 0 ;all themes available to harvest beginning in period 0


AREAS

*A pine 1 172864
*A pine 2 172864
*A pine 3 172864
*A pine 4 172864
*A pine 5 172864
*A pine 6 172864
*A pine 7 172864
*A pine 8 172864
*A pine 9 172864
*A pine 10 172864


CONTROL

*LENGTH 20
*GRAPHICS ON
*REPORTS  ON
*IMAGE    OFF
*BUILD    OFF
*OPTIMIZE OFF
*SCHEDULE ON
*QUEUE    OFF


LANDSCAPE

*THEME covtyp
  pine   pure pine


LIFESPAN

? 50


OPTIMIZE
```

```
*OBJECTIVE
_MAXMIN oldgrowtharea 1.._LENGTH

*CONSTRAINTS
;*****Total volume constraint*****
Totvol = 24062151 1.._LENGTH

;*****Even-flow total volume constraint
_EVEN(totvol) 1.._LENGTH

;*****Non-decline standing inventory*****
_NDY(standinginv) 15.._LENGTH

*FORMAT MOSEK


OUTPUTS

*OUTPUT totvol total volume cut
 *SOURCE harvest tot

*OUTPUT standinginv
 *SOURCE _INVENT tot

*OUTPUT oldgrowtharea
 *SOURCE @AGE(10.._MAXAGE) _INVENT _AREA


REPORTS

*TARGET OG_1_allrep.txt
_ALL 1.._LENGTH

*TARGET surplus.wk1
_ALL   1.._LENGTH

*TARGET ageclassP1.txt
 _AGECLASS 1

*TARGET ogreport.txt
oldgrowtharea 1

*TARGET Harvreport.txt
 TOtvol 1


TRANSITIONS

*CASE _DEATH
 *SOURCE ?
 *TARGET ?  100

*CASE harvest
 *SOURCE ?
 *TARGET ?   100


YIELDS

*Y ?
tot 1 0.139 2.392 10.049 24.108 43.069
      64.454 85.999 106.125 123.964 139.197
      151.856 162.167 170.443 177.012 182.182
      186.226 189.373 191.814 193.701 195.158
      196.281 197.144 197.808 198.318 198.710
      199.011 199.241 199.418 199.554 199.658
```

# Appendix C

# Chapter 3 Modeling System in Python Code

```
# Chapter 3 Modeling system
# Author:  Matt Conrod
# Date: February 20th, 2010

import os

# opening and reading random fire file
import string
population=open("C:\\MC_CHP2_model\\Area_burned_PAFMA_Jan_08.txt","r")
fire_list=map(string.strip, population.readlines())
population.close()

# Defining a "sample with replacement" function to select random fire years
# credit author(s) of random.py
import random

def sample_wr(population, k):
    "Chooses k random elements (with replacement) from a population"
    n = len(population)
    _random, _int = random.random, int  # speed hack
    result = [None] * k
    for i in xrange(k):
        j = _int(_random() * n)
        result[i] = population[j]
    return result

# The draw loop
draw = 1
while draw <=100:


    #refreshing the .are file
    os.system("C:\\MC_CHP2_model\\Refresh_Area.bat")

    print "Draw # "
    print draw

    #The planning period loop
    planning_period = 1
    while planning_period <=20:

        print "Planning Period "
        print planning_period
```

```
#refreshing the .opt file
os.system("C:\\MC_CHP2_model\\Refresh_opt.bat")

#Generating matrix, solving matrix and generating reports
os.system("C:\\MC_CHP2_model\\call_Opt_OG_1.bat")

#**Opening and reading end of period one harvest total
#    report from Woodstock**
f=open("c:\\MC_CHP2_model\\HARVREPORT.TXT","r")
harv_tot=map(string.strip, f.readlines())
f.close()

#deleting text line from start of HARVREPORT.TXT file data
del harv_tot[0:4]
harv_totsplt_temp= [i.split() for i in harv_tot]
del harv_totsplt_temp[1]

#Deleting first variables in HARVREPORT.TXT file so that only
#harvest volume is remaining
harv_totsplt=harv_totsplt_temp[0]
del harv_totsplt[0]
del harv_totsplt[0]
del harv_totsplt[0]
harv_totsplt_str=str(harv_totsplt[0])
harv_totsplt_str_2=harv_totsplt_str.replace(",","")
harv_totsplt_flt=eval(harv_totsplt_str_2)
print 'Current harvest accomlishment: ', harv_totsplt_flt

#Calculating the new harvest request value with the
#"Harv_constraint" value
Opt_str_flt_edit=harv_totsplt_flt
print 'New harvest request: ', Opt_str_flt_edit

#rebuilding the .opt file with new harvest request
f=open("c:\MC_CHP2_model\\OG_1.opt","w")
f.write('; Optimize\n *OBJECTIVE\n _MAX oPVTotalRev - oPVTotalCost
            1.._LENGTH\n \n *CONSTRAINTS\n Totvol <= ')
new_harv=str(Opt_str_flt_edit)
f.write(new_harv)
f.write(' 1.._LENGTH\n oldgrowtharea >= 172864 1.._LENGTH \n _EVEN(totvol)
             1.._LENGTH\n _NDY(standinginv) 15.._LENGTH\n \n *exclude \n burn
             1.._length \n *FORMAT MOSEK')
f.close()

#Generating matrix, solving matrix and generating reports...again...
#with the model III
#values as a constraint
os.system("C:\\MC_CHP2_model\\call_Opt_OG_1.bat")

#****Opening and reading end of period one
#        harvest total report from Woodstock****
f=open("c:\\MC_CHP2_model\\HARVREPORT.TXT","r")
harv_tot=map(string.strip, f.readlines())
f.close()

#deleting text line from start of HARVREPORT.TXT file data
del harv_tot[0:4]
harv_totsplt_temp= [i.split() for i in harv_tot]
del harv_totsplt_temp[1]

#Deleting first variables in HARVREPORT.TXT file so that only harvest
#volume is remaining
harv_totsplt=harv_totsplt_temp[0]
del harv_totsplt[0]
del harv_totsplt[0]
del harv_totsplt[0]
harv_totsplt_str=str(harv_totsplt[0])
harv_totsplt_str_2=harv_totsplt_str.replace(",","")
```

```
harv_totsplt_flt=eval(harv_totsplt_str_2)
print 'Current harvest accomlishment: ', harv_totsplt_flt

#*****Opening and reading end of period one revenue
#          report from Woodstock*****
f=open("c:\\MC_CHP2_model\\revenuereport.TXT","r")
rev_tot=map(string.strip, f.readlines())
f.close()

#deleting text line from start of revenue.TXT file data
del rev_tot[0:4]
rev_totsplt_temp= [i.split() for i in rev_tot]
del rev_totsplt_temp[1]

#Deleting first variables in revenue.TXT file so that only harvest
#volume is remaining
rev_totsplt=rev_totsplt_temp[0]
del rev_totsplt[0]
del rev_totsplt[0]
#del rev_totsplt[0]
rev_totsplt_str=str(rev_totsplt[0])
rev_totsplt_str_2=rev_totsplt_str.replace(",","")
rev_totsplt_flt=eval(rev_totsplt_str_2)
print 'Current revenue accomlishment: ', rev_totsplt_flt


#***Opening and reading end of period one discounted
#      revenue report from Woodstock***
f=open("c:\\MC_CHP2_model\\PVrevenuereport.TXT","r")
PVrev_tot=map(string.strip, f.readlines())
f.close()

#deleting text line from start of revenue.TXT file data
del PVrev_tot[0:4]
PVrev_totsplt_temp= [i.split() for i in PVrev_tot]
del PVrev_totsplt_temp[1]

#Deleting first variables in revenue.TXT file so that only harvest
#volume is remaining
PVrev_totsplt=PVrev_totsplt_temp[0]
del PVrev_totsplt[0]
del PVrev_totsplt[0]
del PVrev_totsplt[0]
del PVrev_totsplt[0]
del PVrev_totsplt[0]
PVrev_totsplt_str=str(PVrev_totsplt[0])
PVrev_totsplt_str_2=PVrev_totsplt_str.replace(",","")
PVrev_totsplt_flt=eval(PVrev_totsplt_str_2)
print 'Current discounted revenue accomlishment: ', PVrev_totsplt_flt

#***Opening and reading end of period one cost report
#      from Woodstock***
f=open("c:\\MC_CHP2_model\\costreport.TXT","r")
cost_tot=map(string.strip, f.readlines())
f.close()

#deleting text line from start of costreport.TXT file data
del cost_tot[0:4]
cost_totsplt_temp= [i.split() for i in cost_tot]
del cost_totsplt_temp[1]

#Deleting first variables in costreport.TXT file so that only harvest
#volume is remaining
cost_totsplt=cost_totsplt_temp[0]
del cost_totsplt[0]
del cost_totsplt[0]
del cost_totsplt[0]
cost_totsplt_str=str(cost_totsplt[0])
```

```
cost_totsplt_str_2=cost_totsplt_str.replace(",","")
cost_totsplt_flt=eval(cost_totsplt_str_2)
print 'Current cost accomlishment: ', cost_totsplt_flt

#***Opening and reading end of period one cost present
#      value report from Woodstock***
f=open("c:\\MC_CHP2_model\\PVcostreport.TXT","r")
PVcost_tot=map(string.strip, f.readlines())
f.close()

#deleting text line from start of PVcost.TXT file data
del PVcost_tot[0:4]
PVcost_totsplt_temp= [i.split() for i in PVcost_tot]
del PVcost_totsplt_temp[1]

#Deleting first variables in PVcost.TXT file so that only harvest
#volume is remaining
PVcost_totsplt=PVcost_totsplt_temp[0]
del PVcost_totsplt[0]
del PVcost_totsplt[0]
del PVcost_totsplt[0]
del PVcost_totsplt[0]
del PVcost_totsplt[0]
PVcost_totsplt_str=str(PVcost_totsplt[0])
PVcost_totsplt_str_2=PVcost_totsplt_str.replace(",","")
PVcost_totsplt_flt=eval(PVcost_totsplt_str_2)
print 'Current discounted cost accomlishment: ', PVcost_totsplt_flt

#***Opening and reading end of period one NPV total report
#      from Woodstock***
f=open("c:\\MC_CHP2_model\\NPV.TXT","r")
NPV_tot=map(string.strip, f.readlines())
f.close()

#deleting text line from start of NPV.TXT file data
del NPV_tot[0:4]
NPV_totsplt_temp= [i.split() for i in NPV_tot]
del NPV_totsplt_temp[1]

#Deleting first variables in oNPV.TXT file so that only harvest
#volume is remaining
NPV_totsplt=NPV_totsplt_temp[0]
del NPV_totsplt[0]
del NPV_totsplt[0]
del NPV_totsplt[0]
NPV_totsplt_str=str(NPV_totsplt[0])
NPV_totsplt_str_2=NPV_totsplt_str.replace(",","")
NPV_totsplt_flt=eval(NPV_totsplt_str_2)
print 'Current NPV accomlishment: ', NPV_totsplt_flt

#***Opening and reading end of period one schedule
#      report from Woodstock***
f=open("c:\\MC_CHP2_model\\SCHEDULE.TXT","r")
SCH_tot=map(string.strip, f.readlines())
f.close()

#deleting text line from start of SCHEDULE.TXT file data
del SCH_tot[0:2]
SCH_totsplt_temp = [i.split() for i in SCH_tot]

#Deleting first variables in schedule.TXT file so that only harvest
#volume is remaining
del SCH_totsplt_temp[0]

#writing the results file - [draw_#] [planning_period] [OG_area]
# [harvest volume]
SCH=open(r'c:\MC_CHP2_model\SCHEDULE_LOG.txt','a')
x=0
```

```
list_length=len(SCH_totsplt_temp)
while x < list_length:
    str_pp=str(planning_period)
    str_draw=str(draw)
    str_SCH_totsplt=str(SCH_totsplt_temp[x])
    SCH.write(str_draw)
    SCH.write(" ")
    SCH.write(str_pp)
    SCH.write(" ")
    SCH.write(str_SCH_totsplt)
    SCH.write(" ")
    SCH.write("\n")
    x=x+1
SCH.close()


# generating 10 random fire years and summing to represent
#total percent area
#burned for the 10-year planning period
periodic_fire = sample_wr((fire_list),10)
print "Randomly drawing annual fire proportions....."
periodic_fire[0]=eval(periodic_fire[0])
periodic_fire[1]=eval(periodic_fire[1])
periodic_fire[2]=eval(periodic_fire[2])
periodic_fire[3]=eval(periodic_fire[3])
periodic_fire[4]=eval(periodic_fire[4])
periodic_fire[5]=eval(periodic_fire[5])
periodic_fire[6]=eval(periodic_fire[6])
periodic_fire[7]=eval(periodic_fire[7])
periodic_fire[8]=eval(periodic_fire[8])
periodic_fire[9]=eval(periodic_fire[9])

total_fire =1-((1-periodic_fire[0])*(1-periodic_fire[1])*(1-periodic_fire[2]) \
               *(1-periodic_fire[3])*(1-periodic_fire[4])*(1-periodic_fire[5]) \
               *(1-periodic_fire[6])*(1-periodic_fire[7])*(1-periodic_fire[8]) \
               *(1-periodic_fire[9]))
print "Summing annual fire proportions for planning period...."
print total_fire

# Opening and reading end of period one age-class report
# from Woodstock run
f=open("C:\\MC_CHP2_model\\ageclassp1.txt","r")
Prd1_AgeC=map(string.strip, f.readlines())
f.close()

# deleting text lines from start of file data
del Prd1_AgeC[0:5]

# Spliting string data into lists
import re
Prd1_AgeC_splt = [i.split() for i in Prd1_AgeC]

# Creating a dictionary (array) indexed by planning period
dict_Prd1_AgeC=dict(Prd1_AgeC_splt)

# Creating a dictionary with area converted to float
dict_Prd1_AgeC_2 = dict(zip(dict_Prd1_AgeC.keys(), ([float(value) for \
         value in dict_Prd1_AgeC.values()])))
#print dict_Prd1_AgeC_2

# Convert keys to integer
dict_Prd1_AgeC_3 = dict(zip(([int(key) for key in dict_Prd1_AgeC_2.keys()]), \
         dict_Prd1_AgeC_2.values()))
#print dict_Prd1_AgeC_3

# calculating burn area
Prd_burn_areas = dict(zip(dict_Prd1_AgeC_3.keys(), \
       ([total_fire*value for value in dict_Prd1_AgeC_3.values()])))
```

```
print "Calculating burn areas...."
#print Prd_burn_areas

# summing burn area
total_burn_area = sum(Prd_burn_areas.values())
print "Summing burn areas..."
print total_burn_area

#calculating new areas
post_burn_areas = dict(zip(dict_Prd1_AgeC_3.keys(), \
       ([(value_1-(total_fire*value_1)) for value_1 in \
       dict_Prd1_AgeC_3.values()])))
print "Calculating new post-burn areas..."
#print post_burn_areas

# Adding burn areas to age-class one
post_burn_areas[1] = post_burn_areas[1]+ total_burn_area
print "Adding burn areas to age-class one...."
#print post_burn_areas

#Generating new .are file post-burn
f=open("c:\\MC_CHP2_model\\OG_1.are","w")
for key in sorted(post_burn_areas):
    f.write('*A pine ''%s %s\n' % (str(key), \
              ''.join(repr(post_burn_areas[key]).strip('[]').split(','))))
f.close()

#Deleting area younger then 90 yrs
AGE_DEL = 1
while AGE_DEL <=9:
    if post_burn_areas.has_key (AGE_DEL):
        del post_burn_areas[AGE_DEL]
    AGE_DEL=AGE_DEL+1

#Calculating old-growth area after burn
og_area=sum(post_burn_areas.values())
print "calculating old-growth areas...."
print og_area

#writing the results file - [draw_#] [planning_period]
#[OG_area] [harvest volume]
og=open(r'c:\MC_CHP2_model\OG_results.txt','a')
str_pp=str(planning_period)
str_draw=str(draw)
str_og_area=str(og_area)
str_harv_totsplt=str(harv_totsplt_flt)
str_rev_totsplt=str(rev_totsplt_flt)
str_PVrev_totsplt=str(PVrev_totsplt_flt)
str_cost_totsplt=str(cost_totsplt_flt)
str_PVcost_totsplt=str(PVcost_totsplt_flt)
str_NPV_totsplt=str(NPV_totsplt_flt)
str_total_fire=str(total_fire)
og.write(str_draw)
og.write(" ")
og.write(str_pp)
og.write(" ")
og.write(str_og_area)
og.write(" ")
og.write(str_harv_totsplt)
og.write(" ")
og.write(str_rev_totsplt)
og.write(" ")
og.write(str_PVrev_totsplt)
og.write(" ")
og.write(str_cost_totsplt)
og.write(" ")
og.write(str_PVcost_totsplt)
og.write(" ")
```

```
    og.write(str_NPV_totsplt)
    og.write(" ")
    og.write(str_total_fire)
    og.write("\n")
    og.close()

    planning_period = planning_period + 1

draw = draw+1
```

# Appendix D

# Chapter 3 Woodstock Model Formulation

```
ACTIONS

*ACTION harvest Y Harvest
  *OPERABLE harvest
    ? _CP >= 1 ;all themes available to harvest beginning in period 0

*ACTION burn n
  *OPERABLE burn
    ? _CP >= 1


AREAS

*A pine 1 172864
*A pine 2 172864
*A pine 3 172864
*A pine 4 172864
*A pine 5 172864
*A pine 6 172864
*A pine 7 172864
*A pine 8 172864
*A pine 9 172864
*A pine 10 172864


CONTROL

 *LENGTH 20
 *GRAPHICS ON
 *REPORTS ON
 *IMAGE OFF
 *BUILD OFF
 *OPTIMIZE OFF
 *SCHEDULE ON
 *QUEUE OFF


LANDSCAPE

*THEME covtyp
  pine   pure pine


LIFESPAN
```

```
? 50


OPTIMIZE

 *OBJECTIVE
 _MAX oPVTotalRev - oPVTotalCost  1.._LENGTH

 *CONSTRAINTS

;****Old forest constraint****
 oldgrowtharea >= 172864 1.._LENGTH

;****Even-flow total volume constraint****
 _EVEN(totvol) 1.._LENGTH

;****Non-declining growing stock constraint****
 _NDY(standinginv) 15.._LENGTH

;****Area burned constraint****
burnarea = Totalarea 1.._LENGTH

*EXCLUDE
;burn 1.._LENGTH

*FORMAT MOSEK


OUTPUTS

*OUTPUT totvol total volume cut
 *SOURCE harvest tot

*OUTPUT standinginv
 *SOURCE _INVENT tot

*OUTPUT burnarea
 *SOURCE burn _AREA

*OUTPUT TOtalArea
 *SOURCE _INVENT _AREA

*OUTPUT oldgrowtharea
 *SOURCE @AGE(10.._MAXAGE) _INVENT _AREA

;****Revenue****

*OUTPUT oTotalRevenue Total revenue
 *SOURCE ? harvest tot * ytot$

*OUTPUT oPVTotalRev Present value of total revenue
 *SOURCE oTotalRevenue * yDisc5%

;****Cost****

*OUTPUT osilvCost Total cost of establisment (site prep, plant)
 *SOURCE ? harvest ysilv$

*OUTPUT oHarvCost Total cost of harvesting
 *SOURCE ? harvest yHarvcost$

*OUTPUT oTotalCost Grand total costs
 *SOURCE osilvCost + oHarvcost

*OUTPUT oPVTotalCost Present value of total costs
 *SOURCE oTotalCost * yDisc5%

;****NPV****
```

```
*OUTPUT oNPV Net Present Value
 *SOURCE oPVTotalrev - oPVtotalcost


REPORTS

*TARGET OG_1_allrep.txt
 _ALL 1.._LENGTH

*TARGET surplus.wk1 {report is sent to 1-2-3 file feb11.wk1}
 _ALL   1.._LENGTH

*TARGET ageclassP1.txt
 _AGECLASS 1

*TARGET ogreport.txt
 oldgrowtharea 1

*TARGET Harvreport.txt
 totvol 1

*TARGET revenuereport.txt
 ototalrevenue 1

*TARGET costreport.txt
 ototalcost 1

*TARGET PVrevenuereport.txt
 oPVtotalrev 1

*TARGET PVcostreport.txt
 oPVtotalcost 1

*TARGET NPV.txt
 oNPV 1

*TARGET SCHEDULE.txt
 _SCHEDULE 1


TRANSITIONS

*CASE _DEATH
 *SOURCE ?
  *TARGET ?  100

*CASE harvest
 *SOURCE ?
  *TARGET ? 100

*CASE BURN
 *SOURCE ?
  *TARGET ? 94  ;94% maintains original age
  *TARGET ? 6 _AGE 0   ; 6% transitions to age 0


YIELDS

;*****Volume Yields******
*Y ?
tot 1 0.139 2.392 10.049 24.108 43.069
      64.454 85.999 106.125 123.964 139.197
      151.856 162.167 170.443 177.012 182.182
      186.226 189.373 191.814 193.701 195.158
      196.281 197.144 197.808 198.318 198.710
      199.011 199.241 199.418 199.554 199.658
```

```
;*****management costs $/ha*****
*YT ?

;*****discount factors*****
yDisc2.5% _DISCOUNTFACTOR(2.5%,10,half)
yDisc5% _DISCOUNTFACTOR(5%,10,half)
yDisc7.5% _DISCOUNTFACTOR(7.5%,10,half)

;*****revenue $/m3*****
ytot$ 1 50

;*****costs $/ha*****
ysilv$ 1 500
yharvcost$ 1 5000
```

# Appendix E

# Chapter 4 Modeling System in Python Code

```
# Chapter 4 Script
# Author:  Matt Conrod
# Date: February 13th, 2010

import os
import time
import random

#opening and reading fire history file
import string
population=open("C:\\MC_CHP3_model\\Area_burned_PAFMA_Jan_08.txt","r")
fire_list=map(string.strip, population.readlines())
population.close()


# Defining a "sample with replacement" function to select random fire years
# credit author(s) of random.py
def sample_wr(population, k):
    "Chooses k random elements (with replacement) from a population"
    n = len(population)
    _random, _int = random.random, int  # speed hack
    result = [None] * k
    for i in xrange(k):
        j = _int(_random() * n)
        result[i] = population[j]
    return result


class Lookup(dict):

    """
    a dictionary which can lookup value by key, or keys by value
    """
    def __init__(self, items=[]):
        """items can be a list of pair_lists or a dictionary"""
        dict.__init__(self, items)

    def get_key(self, value):
        """find the key(s) as a list given a value"""
        return [item[0] for item in self.items() if item[1] == value]

    def get_value(self, key):
        """find the value given a key"""
        return self[key]
```

```
# The draw loop
draw = 1
while draw <=50:


    #refreshing the .are file
    os.system("C:\\MC_CHP3_model\\Refresh_Area.bat")

    print "Draw # "
    print draw

    #The planning period loop
    planning_period = 1
    while planning_period <=20:

        HW_constraint = 22495
        HJP_constraint = 3114
        JPH_constraint = 2422
        HXS_constraint = 5605
        XSH_constraint = 2820
        BS_constraint = 12090
        JP_constraint = 18443
        JPBS_constraint = 12040
        WSBF_constraint = 6216
        TL_constraint = 3423


        #refreshing the .opt file
        os.system("C:\\MC_CHP3_model\\Refresh_opt.bat")


        print "Planning Period "
        print planning_period


        #Generating matrix and solving matrix
        os.system("C:\\MC_CHP3_model\\pre_call_Opt_OG_1.bat")

        exists=0

        #Testing whether a solution is infeasible
        if os.path.exists("c:\\MC_CHP3_model\\Infeasible.msg"):
                exists = 1

        if exists == 1 : #if infeasible - identified by infeasible msg
                print "INFEASIBLE!!!"

                #rebuilding the .opt file with GOALS on old forest constraints!!!
                f=open("c:\MC_CHP3_model\\OG_1.opt","w")
                f.write('; Optimize\n *OBJECTIVE\n _MAX oPVTotalRev - oPVtotalCost -
                          _PENALTY(_ALL) 1.._LENGTH\n \n *CONSTRAINTS\n')
                f.write('\n HWOldgrowtharea >= 22495 1.._LENGTH _GOAL(G1,10000000)')
                f.write('\n HJPOldgrowtharea >= 3114 1.._LENGTH _GOAL(G2,10000000)')
                f.write('\n JPHOldgrowtharea >= 2422 1.._LENGTH _GOAL(G3,10000000)')
                f.write('\n HXSOldgrowtharea >= 5605 1.._LENGTH _GOAL(G4,10000000)')
                f.write('\n XSHOldgrowtharea >= 2820 1.._LENGTH _GOAL(G5,10000000)')
                f.write('\n BSOldgrowtharea >= 12090 1.._LENGTH _GOAL(G6,10000000)')
                f.write('\n JPOldgrowtharea >= 18443 1.._LENGTH _GOAL(G7,10000000)')
                f.write('\n JPBSOldgrowtharea >= 12040 1.._LENGTH _GOAL(G8,10000000)')
                f.write('\n WSBFOldgrowtharea >= 6216 1.._LENGTH _GOAL(G9,10000000)')
                f.write('\n TLOldgrowtharea >= 3423 1.._LENGTH _GOAL(G10,10000000)')
                f.write('\n \n _EVEN(softvol) 1.._LENGTH\n _EVEN(hardvol) 1.._LENGTH\n
                         _NDY(standinginv) 15.._LENGTH\n burnarea = Totalarea 1.._length
                         \n *EXCLUDE \n ;burn 1.._LENGTH \n \n *FORMAT MOSEK')
                f.close()

                #Generating matrix and solving matrix
                os.system("C:\\MC_CHP3_model\\pre_call_Opt_OG_1.bat")
```

```
#Generating reports
os.system("C:\\MC_CHP3_model\\call_Opt_OG_1_reports.bat")



##***************New HW Constraints***********************

#Wiping clean the temp file
erase=open("c:\MC_CHP3_model\OGTEMP.txt","w")
erase.write("")
erase.close()

##Opening HW old growth report and copying to OGTEMP
for line in open("c:\\MC_CHP3_model\\HWOGREPORT.txt","r"):
    if line.startswith("HWOLDGROWTHAREA"):
        P1a=open("c:\MC_CHP3_model\OGTEMP.txt","a")
        P1a.write(line)
        P1a.close()
    else:
        del line

#reading lines of OGTEMP
f=open("c:\\MC_CHP3_model\\OGTEMP.TXT","r")
HWOG=map(string.strip, f.readlines())
f.close()

#spliting lines of OGTEMP
HWOG_temp = [i.split() for i in HWOG]

#removing text from lines
count = 0
while count <= 19:
    HWOG_temp[count].remove("HWOLDGROWTHAREA")
    count = count + 1

#creating dictionary (array) indexed by period
periods = 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20
HWOG=dict(zip(periods, HWOG_temp))
print HWOG

#defining new empty dicitonary
NewHwCont = {}

#populating new dictionary with periods that are less than original constraint
period = 1
while period <= 20:
    HWOG_tempchk = HWOG[period]
    HWOG_tempchk2 = HWOG_tempchk[0].replace(",","")
    HWOG_tempchkflt = eval(HWOG_tempchk2)
    print HWOG_tempchkflt
    if HWOG_tempchkflt < HW_constraint:
        NewHwCont[period] = HWOG_tempchkflt
    period = period + 1

print NewHwCont

##***************New HJP Constraints*********************

#Wiping clean the temp file
erase=open("c:\MC_CHP3_model\OGTEMP.txt","w")
erase.write("")
erase.close()

##Opening HJP old growth report and copying to OGTEMP
for line in open("c:\\MC_CHP3_model\\HJPOGREPORT.txt","r"):
    if line.startswith("HJPOLDGROWTHAREA"):
        P1a=open("c:\MC_CHP3_model\OGTEMP.txt","a")
```

```
        P1a.write(line)
        P1a.close()
    else:
        del line

#reading lines of OGTEMP
f=open("c:\\MC_CHP3_model\\OGTEMP.TXT","r")
HJPOG=map(string.strip, f.readlines())
f.close()

#spliting line of OGTEMP
HJPOG_temp = [i.split() for i in HJPOG]
print HJPOG_temp

#removing text from lines
count = 0
while count <= 19:
    HJPOG_temp[count].remove("HJPOLDGROWTHAREA")
    count = count + 1

#creating dictionary indexed by period
periods = 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20
HJPOG=dict(zip(periods, HJPOG_temp))
print HJPOG

#defining new empty dictionary
NewHJPCont = {}

#Populating new dictionary with periods that are less than original constraint
period = 1
while period <= 20:
    HJPOG_tempchk = HJPOG[period]
    HJPOG_tempchk2 = HJPOG_tempchk[0].replace(",","")
    HJPOG_tempchkflt = eval(HJPOG_tempchk2)
    print HJPOG_tempchkflt
    if HJPOG_tempchkflt < HJP_constraint:
        NewHJPCont[period] = HJPOG_tempchkflt
    period = period + 1


##***************New JPH Constraints**********************

#Wiping clean the temp file
erase=open("c:\MC_CHP3_model\OGTEMP.txt","w")
erase.write("")
erase.close()

##Opening HJP old growth report and copying to OGTEMP
for line in open("c:\\MC_CHP3_model\\JPHOGREPORT.txt","r"):
    if line.startswith("JPHOLDGROWTHAREA"):
        P1a=open("c:\MC_CHP3_model\OGTEMP.txt","a")
        P1a.write(line)
        P1a.close()
    else:
        del line

#reading lines of OGTEMP
f=open("c:\\MC_CHP3_model\\OGTEMP.TXT","r")
JPHOG=map(string.strip, f.readlines())
f.close()
#print JPHOG

#spliting line of OGTEMP
JPHOG_temp = [i.split() for i in JPHOG]
print JPHOG_temp

#removing text from lines
count = 0
```

```
while count <= 19:
    JPHOG_temp[count].remove("JPHOLDGROWTHAREA")
    count = count + 1

#creating dictionary indexed by period
periods = 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20
JPHOG=dict(zip(periods, JPHOG_temp))
print JPHOG

#defining new empty dictionary
NewJPHCont = {}

#Populating new dictionary with periods that are less than original constraint
period = 1
while period <= 20:
    JPHOG_tempchk = JPHOG[period]
    JPHOG_tempchk2 = JPHOG_tempchk[0].replace(",","")
    JPHOG_tempchkflt = eval(JPHOG_tempchk2)
    print JPHOG_tempchkflt
    if JPHOG_tempchkflt < JPH_constraint:
        NewJPHCont[period] = JPHOG_tempchkflt
    period = period + 1

##***************New HXS Constraints*********************

#Wiping clean the temp file
erase=open("c:\MC_CHP3_model\OGTEMP.txt","w")
erase.write("")
erase.close()

##Opening HXS old growth report and copying to OGTEMP
for line in open("c:\\MC_CHP3_model\\HXSOGREPORT.txt","r"):
    if line.startswith("HXSOLDGROWTHAREA"):
        P1a=open("c:\MC_CHP3_model\OGTEMP.txt","a")
        P1a.write(line)
        P1a.close()
    else:
        del line

#reading lines of OGTEMP
f=open("c:\\MC_CHP3_model\\OGTEMP.TXT","r")
HXSOG=map(string.strip, f.readlines())
f.close()
#print HXSOG

#spliting line of OGTEMP
HXSOG_temp = [i.split() for i in HXSOG]
print HXSOG_temp

#removing text from lines
count = 0
while count <= 19:
    HXSOG_temp[count].remove("HXSOLDGROWTHAREA")
    count = count + 1

#creating dictionary indexed by period
periods = 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20
HXSOG=dict(zip(periods, HXSOG_temp))
print HXSOG

#defining new empty dictionary
NewHXSCont = {}

#Populating new dictionary with periods that are less than original constraint
period = 1
while period <= 20:
    HXSOG_tempchk = HXSOG[period]
    HXSOG_tempchk2 = HXSOG_tempchk[0].replace(",","")
```

```
    HXSOG_tempchkflt = eval(HXSOG_tempchk2)
    print HXSOG_tempchkflt
    if HXSOG_tempchkflt < HXS_constraint:
        NewHXSCont[period] = HXSOG_tempchkflt
    period = period + 1

##***************New XSH Constraints********************

#Wiping clean the temp file
erase=open("c:\MC_CHP3_model\OGTEMP.txt","w")
erase.write("")
erase.close()

##Opening HXS old growth report and copying to OGTEMP
for line in open("c:\\MC_CHP3_model\\XSHOGREPORT.txt","r"):
    if line.startswith("XSHOLDGROWTHAREA"):
        P1a=open("c:\MC_CHP3_model\OGTEMP.txt","a")
        P1a.write(line)
        P1a.close()
    else:
        del line

#reading lines of OGTEMP
f=open("c:\\MC_CHP3_model\\OGTEMP.TXT","r")
XSHOG=map(string.strip, f.readlines())
f.close()
#print XSHOG

#spliting line of OGTEMP
XSHOG_temp = [i.split() for i in XSHOG]
print XSHOG_temp

#removing text from lines
count = 0
while count <= 19:
    XSHOG_temp[count].remove("XSHOLDGROWTHAREA")
    count = count + 1

#creating dictionary indexed by period
periods = 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20
XSHOG=dict(zip(periods, XSHOG_temp))
print XSHOG

#defining new empty dictionary
NewXSHCont = {}

#Populating new dictionary with periods that are less than original constraint
period = 1
while period <= 20:
    XSHOG_tempchk = XSHOG[period]
    XSHOG_tempchk2 = XSHOG_tempchk[0].replace(",","")
    XSHOG_tempchkflt = eval(XSHOG_tempchk2)
    print XSHOG_tempchkflt
    if XSHOG_tempchkflt < XSH_constraint:
        NewXSHCont[period] = XSHOG_tempchkflt
    period = period + 1

##***************New BS Constraints*********************

#Wiping clean the temp file
erase=open("c:\MC_CHP3_model\OGTEMP.txt","w")
erase.write("")
erase.close()

##Opening BS old growth report and copying to OGTEMP
for line in open("c:\\MC_CHP3_model\\BSOGREPORT.txt","r"):
    if line.startswith("BSOLDGROWTHAREA"):
        P1a=open("c:\MC_CHP3_model\OGTEMP.txt","a")
```

```
        P1a.write(line)
        P1a.close()
    else:
        del line

#reading lines of OGTEMP
f=open("c:\\MC_CHP3_model\\OGTEMP.TXT","r")
BSOG=map(string.strip, f.readlines())
f.close()
#print BSOG

#spliting line of OGTEMP
BSOG_temp = [i.split() for i in BSOG]
print BSOG_temp

#removing text from lines
count = 0
while count <= 19:
    BSOG_temp[count].remove("BSOLDGROWTHAREA")
    count = count + 1

#creating dictionary indexed by period
periods = 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20
BSOG=dict(zip(periods, BSOG_temp))
print BSOG

#defining new empty dictionary
NewBSCont = {}

#Populating new dictionary with periods that are less than original constraint
period = 1
while period <= 20:
    BSOG_tempchk = BSOG[period]
    BSOG_tempchk2 = BSOG_tempchk[0].replace(",","")
    BSOG_tempchkflt = eval(BSOG_tempchk2)
    print BSOG_tempchkflt
    if BSOG_tempchkflt < BS_constraint:
        NewBSCont[period] = BSOG_tempchkflt
    period = period + 1

##***************New JP Constraints*********************

#Wiping clean the temp file
erase=open("c:\MC_CHP3_model\OGTEMP.txt","w")
erase.write("")
erase.close()

##Opening BS old growth report and copying to OGTEMP
for line in open("c:\\MC_CHP3_model\\JPOGREPORT.txt","r"):
    if line.startswith("JPOLDGROWTHAREA"):
        P1a=open("c:\MC_CHP3_model\OGTEMP.txt","a")
        P1a.write(line)
        P1a.close()
    else:
        del line

#reading lines of OGTEMP
f=open("c:\\MC_CHP3_model\\OGTEMP.TXT","r")
JPOG=map(string.strip, f.readlines())
f.close()
#print BSOG

#spliting line of OGTEMP
JPOG_temp = [i.split() for i in JPOG]
print JPOG_temp

#removing text from lines
count = 0
```

```
while count <= 19:
    JPOG_temp[count].remove("JPOLDGROWTHAREA")
    count = count + 1

#creating dictionary indexed by period
periods = 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20
JPOG=dict(zip(periods, JPOG_temp))
print JPOG

#defining new empty dictionary
NewJPCont = {}

#Populating new dictionary with periods that are less than original constraint
period = 1
while period <= 20:
    JPOG_tempchk = JPOG[period]
    JPOG_tempchk2 = JPOG_tempchk[0].replace(",","")
    JPOG_tempchkflt = eval(JPOG_tempchk2)
    print JPOG_tempchkflt
    if JPOG_tempchkflt < JP_constraint:
        NewJPCont[period] = JPOG_tempchkflt
    period = period + 1

##***************New JPBS Constraints*********************

#Wiping clean the temp file
erase=open("c:\MC_CHP3_model\OGTEMP.txt","w")
erase.write("")
erase.close()

##Opening BS old growth report and copying to OGTEMP
for line in open("c:\\MC_CHP3_model\\JPBSOGREPORT.txt","r"):
    if line.startswith("JPBSOLDGROWTHAREA"):
        P1a=open("c:\MC_CHP3_model\OGTEMP.txt","a")
        P1a.write(line)
        P1a.close()
    else:
        del line

#reading lines of OGTEMP
f=open("c:\\MC_CHP3_model\\OGTEMP.TXT","r")
JPBSOG=map(string.strip, f.readlines())
f.close()

#spliting line of OGTEMP
JPBSOG_temp = [i.split() for i in JPBSOG]
print JPBSOG_temp

#removing text from lines
count = 0
while count <= 19:
    JPBSOG_temp[count].remove("JPBSOLDGROWTHAREA")
    count = count + 1

#creating dictionary indexed by period
periods = 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20
JPBSOG=dict(zip(periods, JPBSOG_temp))
print JPBSOG

#defining new empty dictionary
NewJPBSCont = {}

#Populating new dictionary with periods that are less than original constraint
period = 1
while period <= 20:
    JPBSOG_tempchk = JPBSOG[period]
    JPBSOG_tempchk2 = JPBSOG_tempchk[0].replace(",","")
    JPBSOG_tempchkflt = eval(JPBSOG_tempchk2)
```

```
    print JPBSOG_tempchkflt
    if JPBSOG_tempchkflt < JPBS_constraint:
        NewJPBSCont[period] = JPBSOG_tempchkflt
    period = period + 1

##***************New WSBF Constraints*********************

#Wiping clean the temp file
erase=open("c:\MC_CHP3_model\OGTEMP.txt","w")
erase.write("")
erase.close()

##Opening WSBF old growth report and copying to OGTEMP
for line in open("c:\\MC_CHP3_model\\WSBFOGREPORT.txt","r"):
    if line.startswith("WSBFOLDGROWTHAREA"):
        P1a=open("c:\MC_CHP3_model\OGTEMP.txt","a")
        P1a.write(line)
        P1a.close()
    else:
        del line

#reading lines of OGTEMP
f=open("c:\\MC_CHP3_model\\OGTEMP.TXT","r")
WSBFOG=map(string.strip, f.readlines())
f.close()
#print WSBFOG

#spliting line of OGTEMP
WSBFOG_temp = [i.split() for i in WSBFOG]
print WSBFOG_temp

#removing text from lines
count = 0
while count <= 19:
    WSBFOG_temp[count].remove("WSBFOLDGROWTHAREA")
    count = count + 1

#creating dictionary indexed by period
periods = 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20
WSBFOG=dict(zip(periods, WSBFOG_temp))
print WSBFOG

#defining new empty dictionary
NewWSBFCont = {}

#Populating new dictionary with periods that are less than original constraint
period = 1
while period <= 20:
    WSBFOG_tempchk = WSBFOG[period]
    WSBFOG_tempchk2 = WSBFOG_tempchk[0].replace(",","")
    WSBFOG_tempchkflt = eval(WSBFOG_tempchk2)
    print WSBFOG_tempchkflt
    if WSBFOG_tempchkflt < WSBF_constraint:
        NewWSBFCont[period] = WSBFOG_tempchkflt
    period = period + 1

##***************New TL Constraints*********************

#Wiping clean the temp file
erase=open("c:\MC_CHP3_model\OGTEMP.txt","w")
erase.write("")
erase.close()

##Opening TL old growth report and copying to OGTEMP
for line in open("c:\\MC_CHP3_model\\TLOGREPORT.txt","r"):
    if line.startswith("TLOLDGROWTHAREA"):
        P1a=open("c:\MC_CHP3_model\OGTEMP.txt","a")
        P1a.write(line)
```

```
        P1a.close()
    else:
        del line

#reading lines of OGTEMP
f=open("c:\\MC_CHP3_model\\OGTEMP.TXT","r")
TLOG=map(string.strip, f.readlines())
f.close()
#print TLOG

#spliting line of OGTEMP
TLOG_temp = [i.split() for i in TLOG]
print TLOG_temp

#removing text from lines
count = 0
while count <= 19:
    TLOG_temp[count].remove("TLOLDGROWTHAREA")
    count = count + 1

#creating dictionary indexed by period
periods = 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20
TLOG=dict(zip(periods, TLOG_temp))
print TLOG

#defining new empty dictionary
NewTLCont = {}

#Populating new dictionary with periods that are less than original constraint
period = 1
while period <= 20:
    TLOG_tempchk = TLOG[period]
    TLOG_tempchk2 = TLOG_tempchk[0].replace(",","")
    TLOG_tempchkflt = eval(TLOG_tempchk2)
    print TLOG_tempchkflt
    if TLOG_tempchkflt < TL_constraint:
        NewTLCont[period] = TLOG_tempchkflt
    period = period + 1


#************rebuilding the .opt file with new OG constraints!!!***********

f=open("c:\MC_CHP3_model\\OG_1.opt","w")
f.write('; Optimize\n *OBJECTIVE\n _MAX oPVTotalRev - oPVtotalCost 1.._LENGTH\n
         \n *CONSTRAINTS\n')

#building HW OG constraints
per = 1
while per <= 20:
    if NewHwCont.has_key(per):
        f.write('HWOldgrowtharea >= ')
        print NewHwCont[per]
        NewHWContVal = str((NewHwCont[per])-1)
        print NewHWContVal
        f.write(NewHWContVal)
        f.write(' ')
        NewHWContKey = str(per)
        f.write(NewHWContKey)
        f.write('\n')
    else:
        f.write('HWOldgrowtharea >= ')
        NewHWContVal = str(HW_constraint)
        f.write(NewHWContVal)
        f.write(' ')
        NewHWContKey = str(per)
        f.write(NewHWContKey)
        f.write('\n')
    per = per + 1
```

```
#building HJP OG constraints
per = 1
while per <= 20:
    if NewHJPCont.has_key(per):
        f.write('HJPOldgrowtharea >= ')
        NewHJPContVal = str((NewHJPCont[per])-1)
        f.write(NewHJPContVal)
        f.write(' ')
        NewHJPContKey = str(per)
        f.write(NewHJPContKey)
        f.write('\n')
    else:
        f.write('HJPOldgrowtharea >= ')
        NewHJPContVal = str(HJP_constraint)
        f.write(NewHJPContVal)
        f.write(' ')
        NewHJPContKey = str(per)
        f.write(NewHJPContKey)
        f.write('\n')
    per = per + 1

#building JPH OG constraints
per = 1
while per <= 20:
    if NewJPHCont.has_key(per):
        f.write('JPHOldgrowtharea >= ')
        NewJPHContVal = str((NewJPHCont[per])-1)
        f.write(NewJPHContVal)
        f.write(' ')
        NewJPHContKey = str(per)
        f.write(NewJPHContKey)
        f.write('\n')
    else:
        f.write('JPHOldgrowtharea >= ')
        NewJPHContVal = str(JPH_constraint)
        f.write(NewJPHContVal)
        f.write(' ')
        NewJPHContKey = str(per)
        f.write(NewJPHContKey)
        f.write('\n')
    per = per + 1

#building HXS OG constraints
per = 1
while per <= 20:
    if NewHXSCont.has_key(per):
        f.write('HXSOldgrowtharea >= ')
        NewHXSContVal = str((NewHXSCont[per])-1)
        f.write(NewHXSContVal)
        f.write(' ')
        NewHXSContKey = str(per)
        f.write(NewHXSContKey)
        f.write('\n')
    else:
        f.write('HXSOldgrowtharea >= ')
        NewHXSContVal = str(HXS_constraint)
        f.write(NewHXSContVal)
        f.write(' ')
        NewHXSContKey = str(per)
        f.write(NewHXSContKey)
        f.write('\n')
    per = per + 1

#building XSH OG constraints
per = 1
while per <= 20:
    if NewXSHCont.has_key(per):
```

```
        f.write('XSHOldgrowtharea >= ')
        NewXSHContVal = str((NewXSHCont[per])-1)
        f.write(NewXSHContVal)
        f.write(' ')
        NewXSHContKey = str(per)
        f.write(NewXSHContKey)
        f.write('\n')
    else:
        f.write('XSHOldgrowtharea >= ')
        NewXSHContVal = str(XSH_constraint)
        f.write(NewXSHContVal)
        f.write(' ')
        NewXSHContKey = str(per)
        f.write(NewXSHContKey)
        f.write('\n')
    per = per + 1

#building BS OG constraints
per = 1
while per <= 20:
    if NewBSCont.has_key(per):
        f.write('BSOldgrowtharea >= ')
        NewBSContVal = str((NewBSCont[per])-1)
        f.write(NewBSContVal)
        f.write(' ')
        NewBSContKey = str(per)
        f.write(NewBSContKey)
        f.write('\n')
    else:
        f.write('BSOldgrowtharea >= ')
        NewBSContVal = str(BS_constraint)
        f.write(NewBSContVal)
        f.write(' ')
        NewBSContKey = str(per)
        f.write(NewBSContKey)
        f.write('\n')
    per = per + 1

#building JP OG constraints
per = 1
while per <= 20:
    if NewJPCont.has_key(per):
        f.write('JPOldgrowtharea >= ')
        NewjpContVal = str((NewJPCont[per])-1)
        f.write(NewJPContVal)
        f.write(' ')
        NewJPContKey = str(per)
        f.write(NewJPContKey)
        f.write('\n')
    else:
        f.write('JPOldgrowtharea >= ')
        NewJPContVal = str(JP_constraint)
        f.write(NewJPContVal)
        f.write(' ')
        NewJPContKey = str(per)
        f.write(NewJPContKey)
        f.write('\n')
    per = per + 1

#building JPBS OG constraints
per = 1
while per <= 20:
    if NewJPBSCont.has_key(per):
        f.write('JPBSOldgrowtharea >= ')
        NewjpBSContVal = str((NewJPBSCont[per])-1)
        f.write(NewJPBSContVal)
        f.write(' ')
        NewJPBSContKey = str(per)
```

```
                f.write(NewJPBSContKey)
                f.write('\n')
            else:
                f.write('JPBSOldgrowtharea >= ')
                NewJPBSContVal = str(JPBS_constraint)
                f.write(NewJPBSContVal)
                f.write(' ')
                NewJPBSContKey = str(per)
                f.write(NewJPBSContKey)
                f.write('\n')
            per = per + 1

        #building WSBF OG constraints
        per = 1
        while per <= 20:
            if NewWSBFCont.has_key(per):
                f.write('WSBFOldgrowtharea >= ')
                NewWSBFContVal = str((NewWSBFCont[per]-1))
                f.write(NewWSBFContVal)
                f.write(' ')
                NewWSBFContKey = str(per)
                f.write(NewWSBFContKey)
                f.write('\n')
            else:
                f.write('WSBFOldgrowtharea >= ')
                NewWSBFContVal = str(WSBF_constraint)
                f.write(NewWSBFContVal)
                f.write(' ')
                NewWSBFContKey = str(per)
                f.write(NewWSBFContKey)
                f.write('\n')
            per = per + 1

        #building TL OG constraints
        per = 1
        while per <= 20:
            if NewTLCont.has_key(per):
                f.write('TLOldgrowtharea >= ')
                NewTLContVal = str((NewTLCont[per])-1)
                f.write(NewTLContVal)
                f.write(' ')
                NewTLContKey = str(per)
                f.write(NewTLContKey)
                f.write('\n')
            else:
                f.write('TLOldgrowtharea >= ')
                NewTLContVal = str(TL_constraint)
                f.write(NewTLContVal)
                f.write(' ')
                NewTLContKey = str(per)
                f.write(NewTLContKey)
                f.write('\n')
            per = per + 1

        f.write('\n \n _EVEN(softvol) 1.._LENGTH\n \n _EVEN(hardvol) 1.._LENGTH\n
                 _NDY(standinginv) 15.._LENGTH\n burnarea = Totalarea 1.._length \n *EXCLUDE
                 \n ;burn 1.._LENGTH \n \n *FORMAT MOSEK')
        f.close()

#Making a copy of the new opt section
os.system("C:\\MC_CHP3_model\\backupOPT.bat")

#Generating matrix and solving matrix
os.system("C:\\MC_CHP3_model\\pre_call_Opt_OG_1.bat")

#Generating reports
os.system("C:\\MC_CHP3_model\\call_Opt_OG_1_reports.bat")
```

```
#***Opening and reading end of period one harvest total report from Woodstock***
f=open("c:\\MC_CHP3_model\\HARVREPORT.TXT","r")
harv_tot=map(string.strip, f.readlines())
f.close()

#deleting text line from start of HARVREPORT.TXT file data
del harv_tot[0:4]
harv_totsplt_temp= [i.split() for i in harv_tot]
del harv_totsplt_temp[1]

#Deleting first variables in HARVREPORT.TXT file so that only harvest volume is remaining
harv_totsplt=harv_totsplt_temp[0]
del harv_totsplt[0]
del harv_totsplt[0]
del harv_totsplt[0]
harv_totsplt_str=str(harv_totsplt[0])
harv_totsplt_str_2=harv_totsplt_str.replace(",","")
harv_totsplt_flt=eval(harv_totsplt_str_2)
print 'Current harvest accomlishment: ', harv_totsplt_flt

#Calculating the new harvest request value with the "Harv_constraint" value
Opt_str_flt_edit=harv_totsplt_flt
print 'New harvest request: ', Opt_str_flt_edit

#rebuilding the .opt file with new harvest request set as a <= constraint and
#the new OG constraints
f=open("c:\MC_CHP3_model\\OG_1.opt","w")
f.write('; Optimize\n *OBJECTIVE\n _MAX oPVTotalRev - oPVTotalCost  1.._LENGTH\n
           \n *CONSTRAINTS\n Totvol <= ')
new_harv=str(Opt_str_flt_edit)
f.write(new_harv)
f.write(' 1.._LENGTH \n')
for line in open("c:\\MC_CHP3_model\\backupOG_1.opt","r"):
    print line
    if line.startswith("HWOldgrowtharea"):
        f.write(line)
    else:
        del line

for line in open("c:\\MC_CHP3_model\\backupOG_1.opt","r"):
    print line
    if line.startswith("HJPOldgrowtharea"):
        f.write(line)
    else:
        del line

for line in open("c:\\MC_CHP3_model\\backupOG_1.opt","r"):
    print line
    if line.startswith("JPHOldgrowtharea"):
        f.write(line)
    else:
        del line

for line in open("c:\\MC_CHP3_model\\backupOG_1.opt","r"):
    print line
    if line.startswith("HXSOldgrowtharea"):
        f.write(line)
    else:
        del line

for line in open("c:\\MC_CHP3_model\\backupOG_1.opt","r"):
    print line
    if line.startswith("XSHOldgrowtharea"):
        f.write(line)
    else:
        del line

for line in open("c:\\MC_CHP3_model\\backupOG_1.opt","r"):
```

```
    print line
    if line.startswith("BSOldgrowtharea"):
        f.write(line)
    else:
        del line

for line in open("c:\\MC_CHP3_model\\backupOG_1.opt","r"):
    print line
    if line.startswith("JPOldgrowtharea"):
        f.write(line)
    else:
        del line

for line in open("c:\\MC_CHP3_model\\backupOG_1.opt","r"):
    print line
    if line.startswith("JPBSOldgrowtharea"):
        f.write(line)
    else:
        del line

for line in open("c:\\MC_CHP3_model\\backupOG_1.opt","r"):
    print line
    if line.startswith("WSBFOldgrowtharea"):
        f.write(line)
    else:
        del line

for line in open("c:\\MC_CHP3_model\\backupOG_1.opt","r"):
    print line
    if line.startswith("TLOldgrowtharea"):
        f.write(line)
    else:
        del line


f.write('\n _EVEN(softvol) 1.._LENGTH\n _EVEN(hardvol) 1.._LENGTH\n _NDY(standinginv)
           15.._LENGTH\n \n *exclude \n burn 1.._length \n *FORMAT MOSEK')
f.close()

os.system("C:\\MC_CHP3_model\\call_Opt_OG_1_1.bat") #generating matrix
os.system("C:\\MC_CHP3_model\\OG_1_Gen.bat") #solving matrix
os.system("C:\\MC_CHP3_model\\call_Opt_OG_1_3.bat") #generating reports

print "second optimization"

#************READING FINAL REPORTS**************
#**Opening and reading end of period one harvest total report from Woodstock**
f=open("c:\\MC_CHP3_model\\HARVREPORT.TXT","r")
harv_tot=map(string.strip, f.readlines())
f.close()

#deleting text line from start of HARVREPORT.TXT file data
del harv_tot[0:4]
harv_totsplt_temp= [i.split() for i in harv_tot]
del harv_totsplt_temp[1]

#Deleting first variables in HARVREPORT.TXT file so that only harvest volume is remaining
harv_totsplt=harv_totsplt_temp[0]
del harv_totsplt[0]
del harv_totsplt[0]
del harv_totsplt[0]
harv_totsplt_str=str(harv_totsplt[0])
harv_totsplt_str_2=harv_totsplt_str.replace(",","")
harv_totsplt_flt=eval(harv_totsplt_str_2)
print 'Current harvest accomlishment: ', harv_totsplt_flt

#**Opening and reading end of period one revenue report from Woodstock**
f=open("c:\\MC_CHP3_model\\revenuereport.TXT","r")
```

```
rev_tot=map(string.strip, f.readlines())
f.close()

#deleting text line from start of revenue.TXT file data
del rev_tot[0:4]
rev_totsplt_temp= [i.split() for i in rev_tot]
del rev_totsplt_temp[1]

#Deleting first variables in revenue.TXT file so that only harvest volume is remaining
rev_totsplt=rev_totsplt_temp[0]
del rev_totsplt[0]
del rev_totsplt[0]
#del rev_totsplt[0]
rev_totsplt_str=str(rev_totsplt[0])
rev_totsplt_str_2=rev_totsplt_str.replace(",","")
rev_totsplt_flt=eval(rev_totsplt_str_2)
print 'Current revenue accomlishment: ', rev_totsplt_flt


#**Opening and reading end of period one discounted revenue report from Woodstock**
f=open("c:\\MC_CHP3_model\\PVrevenuereport.TXT","r")
PVrev_tot=map(string.strip, f.readlines())
f.close()

#deleting text line from start of revenue.TXT file data
del PVrev_tot[0:4]
PVrev_totsplt_temp= [i.split() for i in PVrev_tot]
del PVrev_totsplt_temp[1]

#Deleting first variables in revenue.TXT file so that only harvest volume is remaining
PVrev_totsplt=PVrev_totsplt_temp[0]
del PVrev_totsplt[0]
del PVrev_totsplt[0]
del PVrev_totsplt[0]
del PVrev_totsplt[0]
del PVrev_totsplt[0]
PVrev_totsplt_str=str(PVrev_totsplt[0])
PVrev_totsplt_str_2=PVrev_totsplt_str.replace(",","")
PVrev_totsplt_flt=eval(PVrev_totsplt_str_2)
print 'Current discounted revenue accomlishment: ', PVrev_totsplt_flt

#**Opening and reading end of period one cost report from Woodstock**
f=open("c:\\MC_CHP3_model\\costreport.TXT","r")
cost_tot=map(string.strip, f.readlines())
f.close()

#deleting text line from start of costreport.TXT file data
del cost_tot[0:4]
cost_totsplt_temp= [i.split() for i in cost_tot]
del cost_totsplt_temp[1]

#Deleting first variables in costreport.TXT file so that only harvest volume is remaining
cost_totsplt=cost_totsplt_temp[0]
del cost_totsplt[0]
del cost_totsplt[0]
del cost_totsplt[0]
cost_totsplt_str=str(cost_totsplt[0])
cost_totsplt_str_2=cost_totsplt_str.replace(",","")
cost_totsplt_flt=eval(cost_totsplt_str_2)
print 'Current cost accomlishment: ', cost_totsplt_flt

#**Opening and reading end of period one cost present value report from Woodstock**
f=open("c:\\MC_CHP3_model\\PVcostreport.TXT","r")
PVcost_tot=map(string.strip, f.readlines())
f.close()

#deleting text line from start of PVcost.TXT file data
del PVcost_tot[0:4]
```

```
PVcost_totsplt_temp= [i.split() for i in PVcost_tot]
del PVcost_totsplt_temp[1]

#Deleting first variables in PVcost.TXT file so that only harvest volume is remaining
PVcost_totsplt=PVcost_totsplt_temp[0]
del PVcost_totsplt[0]
del PVcost_totsplt[0]
del PVcost_totsplt[0]
del PVcost_totsplt[0]
del PVcost_totsplt[0]
PVcost_totsplt_str=str(PVcost_totsplt[0])
PVcost_totsplt_str_2=PVcost_totsplt_str.replace(",","")
PVcost_totsplt_flt=eval(PVcost_totsplt_str_2)
print 'Current discounted cost accomlishment: ', PVcost_totsplt_flt

#***Opening and reading end of period one NPV total report from Woodstock***
f=open("c:\\MC_CHP3_model\\NPV.TXT","r")
NPV_tot=map(string.strip, f.readlines())
f.close()

#deleting text line from start of NPV.TXT file data
del NPV_tot[0:4]
NPV_totsplt_temp= [i.split() for i in NPV_tot]
del NPV_totsplt_temp[1]

#Deleting first variables in oNPV.TXT file so that only harvest volume is remaining
NPV_totsplt=NPV_totsplt_temp[0]
del NPV_totsplt[0]
del NPV_totsplt[0]
del NPV_totsplt[0]
NPV_totsplt_str=str(NPV_totsplt[0])
NPV_totsplt_str_2=NPV_totsplt_str.replace(",","")
NPV_totsplt_flt=eval(NPV_totsplt_str_2)
print 'Current NPV accomlishment: ', NPV_totsplt_flt

#***Opening and reading end of period one schedule report from Woodstock***
f=open("c:\\MC_CHP3_model\\SCHEDULE.TXT","r")
SCH_tot=map(string.strip, f.readlines())
f.close()

#deleting text line from start of SCHEDULE.TXT file data
del SCH_tot[0:2]
SCH_totsplt_temp = [i.split() for i in SCH_tot]

#Deleting first variables in schedule.TXT file so that only harvest volume is remaining
del SCH_totsplt_temp[0]

#writing the a schedule log file - [draw_#] [planning_period] [OG_area] [harvest volume]
SCH=open(r'c:\MC_CHP3_model\SCHEDULE_LOG.txt','a')
x=0
list_length=len(SCH_totsplt_temp)
while x < list_length:
    str_pp=str(planning_period)
    str_draw=str(draw)
    str_SCH_totsplt=str(SCH_totsplt_temp[x])
    SCH.write(str_draw)
    SCH.write(" ")
    SCH.write(str_pp)
    SCH.write(" ")
    SCH.write(str_SCH_totsplt)
    SCH.write(" ")
    SCH.write("\n")
    x=x+1
SCH.close()


# generating 10 random fire years and summing to represent total percent area
#burned for the 10-year planning period
```

```
periodic_fire = sample_wr((fire_list),10)
print "Randomly drawing annual fire proportions....."
periodic_fire[0]=eval(periodic_fire[0])
periodic_fire[1]=eval(periodic_fire[1])
periodic_fire[2]=eval(periodic_fire[2])
periodic_fire[3]=eval(periodic_fire[3])
periodic_fire[4]=eval(periodic_fire[4])
periodic_fire[5]=eval(periodic_fire[5])
periodic_fire[6]=eval(periodic_fire[6])
periodic_fire[7]=eval(periodic_fire[7])
periodic_fire[8]=eval(periodic_fire[8])
periodic_fire[9]=eval(periodic_fire[9])


##calculating periodic fire
total_fire =1-((1-periodic_fire[0])*(1-periodic_fire[1])*(1-periodic_fire[2]) \
                    *(1-periodic_fire[3])*(1-periodic_fire[4])*(1-periodic_fire[5]) \
                    *(1-periodic_fire[6])*(1-periodic_fire[7])*(1-periodic_fire[8]) \
                    *(1-periodic_fire[9]))
print "Summing annual fire proportions for planning period...."
print total_fire

adjusted_fire=total_fire

##wiping temp ageclass files clean
P1a=open("c:\MC_CHP3_model\AgeclassP1a.txt","w")
P1a.write("")
P1a.close()

P1b=open("c:\MC_CHP3_model\AgeclassP1b.txt","w")
P1b.write("")
P1b.close()

P1c=open("c:\MC_CHP3_model\AgeclassP1c.txt","w")
P1c.write("")
P1c.close()


##**Breaking up the ageclassP1 files into three separate files for fire update procedure**

##Breaking off the first six development types from the AgeclassP1 file
for line in open("c:\\MC_CHP3_model\\ageclassp1.txt","r"):
#fh = open("c:\\MC_CHP3_model\\ageclassp1.txt","r")
#for line in fh.readline():
    if line.startswith(" Period        XSH1"):
        break
    else:
        P1a=open("c:\MC_CHP3_model\AgeclassP1a.txt","a")
        P1a.write(line)
        P1a.close()

##erasing the first five text lines from the first temp ageclass file
f=open("c:\\MC_CHP3_model\\ageclassp1a.txt","r")
agedata = f.readlines()

f.close()

del agedata[0:5]

f=open("c:\\MC_CHP3_model\\ageclassp1a.txt","w")
f.writelines(agedata)
f.close()


##Breaking off the second set of six development types from the AgeclassP1 file
fh = open("c:\\MC_CHP3_model\\ageclassp1.txt","r")
lines = fh.readlines()
counter=0
```

```
for line in lines:
    counter=counter + 1
    #print counter
    if " Period        XSH1" not in line:
        continue
    else:
        break
del lines[0:counter]
#print lines
P1b=open("c:\MC_CHP3_model\AgeclassP1b.txt","a")
#linesSTR = str(lines)
P1b.writelines(lines)
P1b.close()
fh.close()

for line in open("c:\\MC_CHP3_model\\ageclassP1b.txt","r"):
    if line.startswith(" Period       JPBS1"):
        break
    else:
        P1c=open("c:\MC_CHP3_model\AgeclassP1c.txt","a")
        P1c.write(line)
        P1c.close()

##Breaking off the last five development types from the ageclassP1 file
fh = open("c:\\MC_CHP3_model\\ageclassp1.txt","r")
lines = fh.readlines()
counter=0
for line in lines:
    counter=counter + 1
    #print counter
    if " Period       JPBS1" not in line:
        continue
    else:
        break
del lines[0:counter]
#print lines
P1b=open("c:\MC_CHP3_model\AgeclassP1b.txt","w")
P1b.writelines(lines)
P1b.close()
fh.close()


#**pply fire update procedure on the empty age class files and re-writing the area file**

##Update of first six development types
f=open("C:\\MC_CHP3_model\\ageclassp1a.txt","r")
prd = []; D1 = []; D2 = []; D3 = []; D4 = []; D5 = []; D6 = [];
for line in f:
    prdp, DD1, DD2, DD3, DD4, DD5, DD6 = line.split()
    prd.append(int(prdp))
    D1.append(float(DD1))
    D2.append(float(DD2))
    D3.append(float(DD3))
    D4.append(float(DD4))
    D5.append(float(DD5))
    D6.append(float(DD6))
f.close()


DT1=dict(zip(prd, D1))
DT2=dict(zip(prd, D2))
DT3=dict(zip(prd, D3))
DT4=dict(zip(prd, D4))
DT5=dict(zip(prd, D5))
DT6=dict(zip(prd, D6))

##
##*****************DT1**********************
```

```
## calculating burn area DT1
burn_areas_DT1 = dict(zip(DT1.keys(), \
         ([adjusted_fire*value for value in DT1.values()])))
print "Calculating burn areas for DT1...."
#print burn_areas_DT1

# summing burn area DT1
total_burn_area_DT1 = sum(burn_areas_DT1.values())
print "Summing burn areas..."
#print total_burn_area_DT1

#calculating new areas DT1
post_burn_areas_DT1 = dict(zip(DT1.keys(), ([(value_1-(adjusted_fire*value_1)) \
         for value_1 in DT1.values()])))
print "Calculating new post-burn areas..."
#print post_burn_areas_DT1

# Adding burn areas to age-class one DT1
post_burn_areas_DT1[1] = post_burn_areas_DT1[1]+ total_burn_area_DT1
print "Adding burn areas to age-class one...."
#print post_burn_areas_DT1

zero =0.0
look = Lookup(post_burn_areas_DT1)
deletlers = look.get_key(zero)
print deletlers

while deletlers:
    del post_burn_areas_DT1[deletlers.pop()]

#Generating new .are file post-burn
f=open("c:\\MC_CHP3_model\\OG_1.are","w")
for key in sorted(post_burn_areas_DT1):
    f.write('*A HW1 ''%s %s\n' % (str(key), \
              ''.join(repr(post_burn_areas_DT1[key]).strip('[]').split(','))))
f.close()

#Deleting area younger then 90 yrs
AGE_DEL = 1
while AGE_DEL <=7:
    if post_burn_areas_DT1.has_key (AGE_DEL):
        del post_burn_areas_DT1[AGE_DEL]
    AGE_DEL=AGE_DEL+1

#Calculating old-growth area after burn
HWOG_sum=sum(post_burn_areas_DT1.values())
print "calculating HW old-growth areas so far...."
print HWOG_sum

##*****************DT2**********************
## calculating burn area DT2
burn_areas_DT2 = dict(zip(DT2.keys(), \
         ([adjusted_fire*value for value in DT2.values()])))
print "Calculating burn areas for DT2...."
#print burn_areas_DT2

# summing burn area DT2
total_burn_area_DT2 = sum(burn_areas_DT2.values())
print "Summing burn areas..."
#print total_burn_area_DT2

#calculating new areas DT2
post_burn_areas_DT2 = dict(zip(DT2.keys(), ([(value_1-(adjusted_fire*value_1)) \
          for value_1 in DT2.values()])))
print "Calculating new post-burn areas..."
#print post_burn_areas_DT2

# Adding burn areas to age-class one DT2
```

```
post_burn_areas_DT2[1] = post_burn_areas_DT2[1]+ total_burn_area_DT2
print "Adding burn areas to age-class one...."
#print post_burn_areas_DT2

zero =0.0
look = Lookup(post_burn_areas_DT2)
deletlers = look.get_key(zero)
print deletlers

while deletlers:
    del post_burn_areas_DT2[deletlers.pop()]

#Generating new .are file post-burn
f=open("c:\\MC_CHP3_model\\OG_1.are","a")
for key in sorted(post_burn_areas_DT2):
    f.write('*A HW2 ''%s %s\n' % (str(key), \
                 ''.join(repr(post_burn_areas_DT2[key]).strip('[]').split(','))))
f.close()

#Deleting area younger then 90 yrs
AGE_DEL = 1
while AGE_DEL <=7:
    if post_burn_areas_DT2.has_key (AGE_DEL):
        del post_burn_areas_DT2[AGE_DEL]
    AGE_DEL=AGE_DEL+1

#Calculating old-growth area after burn
og_area_DT2=sum(post_burn_areas_DT2.values())
print "calculating HW2 old-growth areas...."
print og_area_DT2
print "total HW old-growth so far..."
HWOG_sum=HWOG_sum + og_area_DT2
print HWOG_sum


##*****************DT3**********************
##calculating burn area DT3
burn_areas_DT3 = dict(zip(DT3.keys(), \
         ([adjusted_fire*value for value in DT3.values()])))
print "Calculating burn areas for DT3...."
#print burn_areas_DT3

#summing burn area DT3
total_burn_area_DT3 = sum(burn_areas_DT3.values())
print "Summing burn areas..."
#print total_burn_area_DT3

#calculating new areas DT3
post_burn_areas_DT3 = dict(zip(DT3.keys(), ([(value_1-(adjusted_fire*value_1)) \
          for value_1 in DT3.values()])))
print "Calculating new post-burn areas..."
#print post_burn_areas

# Adding burn areas to age-class one DT3
post_burn_areas_DT3[1] = post_burn_areas_DT3[1]+ total_burn_area_DT3
print "Adding burn areas to age-class one...."
#print post_burn_areas_DT3

zero =0.0
look = Lookup(post_burn_areas_DT3)
deletlers = look.get_key(zero)
print deletlers

while deletlers:
    del post_burn_areas_DT3[deletlers.pop()]

#Generating new .are file post-burn
f=open("c:\\MC_CHP3_model\\OG_1.are","a")
```

```
for key in sorted(post_burn_areas_DT3):
    f.write('*A HJP ''%s %s\n' % (str(key),
              ''.join(repr(post_burn_areas_DT3[key]).strip('[]').split(','))))
f.close()

#Deleting area younger then 90 yrs
AGE_DEL = 1
while AGE_DEL <=7:
    if post_burn_areas_DT3.has_key (AGE_DEL):
        del post_burn_areas_DT3[AGE_DEL]
    AGE_DEL=AGE_DEL+1

#Calculating old-growth area after burn
og_area_DT3=sum(post_burn_areas_DT3.values())
print "calculating HJP old-growth areas...."
print og_area_DT3
HJPOG_sum=og_area_DT3
print "Total HJP old-growth so far..."
print HJPOG_sum

##*****************DT4***********************
##calculating burn area DT4
burn_areas_DT4 = dict(zip(DT4.keys(), \
         ([adjusted_fire*value for value in DT4.values()])))
print "Calculating burn areas for DT4...."
#print burn_areas_DT4

#summing burn area DT4
total_burn_area_DT4 = sum(burn_areas_DT4.values())
print "Summing burn areas..."
#print total_burn_area_DT4

#calculating new areas DT4
post_burn_areas_DT4 = dict(zip(DT4.keys(), ([(value_1-(adjusted_fire*value_1)) \
           for value_1 in DT4.values()])))
print "Calculating new post-burn areas..."
#print post_burn_areas

# Adding burn areas to age-class one DT4
post_burn_areas_DT4[1] = post_burn_areas_DT4[1]+ total_burn_area_DT4
print "Adding burn areas to age-class one...."
#print post_burn_areas_DT4

zero =0.0
look = Lookup(post_burn_areas_DT4)
deletlers = look.get_key(zero)
print deletlers

while deletlers:
    del post_burn_areas_DT4[deletlers.pop()]

#Generating new .are file post-burn
f=open("c:\\MC_CHP3_model\\OG_1.are","a")
for key in sorted(post_burn_areas_DT4):
    f.write('*A JPH ''%s %s\n' % (str(key), \
               ''.join(repr(post_burn_areas_DT4[key]).strip('[]').split(','))))
f.close()

#Deleting area younger then 90 yrs
AGE_DEL = 1
while AGE_DEL <=7:
    if post_burn_areas_DT4.has_key (AGE_DEL):
        del post_burn_areas_DT4[AGE_DEL]
    AGE_DEL=AGE_DEL+1

#Calculating old-growth area after burn
og_area_DT4=sum(post_burn_areas_DT4.values())
print "calculating JPH old-growth areas...."
```

```
print og_area_DT4
JPHOG_sum=og_area_DT4
print "Total JPH old-growth so far..."
print JPHOG_sum

##*****************DT5**********************
##calculating burn area DT5
burn_areas_DT5 = dict(zip(DT5.keys(), \
         ([adjusted_fire*value for value in DT5.values()])))
print "Calculating burn areas for DT5...."
#print burn_areas_DT5

#summing burn area DT5
total_burn_area_DT5 = sum(burn_areas_DT5.values())
print "Summing burn areas..."
#print total_burn_area_DT5

#calculating new areas DT5
post_burn_areas_DT5 = dict(zip(DT5.keys(), \
          ([(value_1-(adjusted_fire*value_1)) for value_1 in DT5.values()])))
print "Calculating new post-burn areas..."
#print post_burn_areas

# Adding burn areas to age-class one DT5
post_burn_areas_DT5[1] = post_burn_areas_DT5[1]+ total_burn_area_DT5
print "Adding burn areas to age-class one...."
#print post_burn_areas_DT5

zero =0.0
look = Lookup(post_burn_areas_DT5)
deletlers = look.get_key(zero)
print deletlers

while deletlers:
    del post_burn_areas_DT5[deletlers.pop()]


#Generating new .are file post-burn
f=open("c:\\MC_CHP3_model\\OG_1.are","a")
for key in sorted(post_burn_areas_DT5):
    f.write('*A HXS1 ''%s %s\n' % (str(key), \
             ''.join(repr(post_burn_areas_DT5[key]).strip('[]').split(','))))
f.close()

#Deleting area younger then 90 yrs
AGE_DEL = 1
while AGE_DEL <=7:
    if post_burn_areas_DT5.has_key (AGE_DEL):
        del post_burn_areas_DT5[AGE_DEL]
    AGE_DEL=AGE_DEL+1

#Calculating old-growth area after burn
og_area_DT5=sum(post_burn_areas_DT5.values())
print "calculating HXS1 old-growth areas...."
print og_area_DT5
HXSOG_sum=og_area_DT5
print "Total HXS old-growth so far..."
print HXSOG_sum

##*****************DT6**********************
##calculating burn area DT6
burn_areas_DT6 = dict(zip(DT6.keys(), \
        ([adjusted_fire*value for value in DT6.values()])))
print "Calculating burn areas for DT6...."
#print burn_areas_DT6

#summing burn area DT6
total_burn_area_DT6 = sum(burn_areas_DT6.values())
```

```
print "Summing burn areas..."
#print total_burn_area_DT6

#calculating new areas DT6
post_burn_areas_DT6 = dict(zip(DT6.keys(), \
          ([(value_1-(adjusted_fire*value_1)) for value_1 in DT6.values()])))
print "Calculating new post-burn areas..."
#print post_burn_areas

# Adding burn areas to age-class one DT6
post_burn_areas_DT6[1] = post_burn_areas_DT6[1]+ total_burn_area_DT6
print "Adding burn areas to age-class one...."
#print post_burn_areas_DT6

zero =0.0
look = Lookup(post_burn_areas_DT6)
deletlers = look.get_key(zero)
print deletlers

while deletlers:
    del post_burn_areas_DT6[deletlers.pop()]


#Generating new .are file post-burn
f=open("c:\\MC_CHP3_model\\OG_1.are","a")
for key in sorted(post_burn_areas_DT6):
    f.write('*A HXS2 ''%s %s\n' % (str(key), \
              ''.join(repr(post_burn_areas_DT6[key]).strip('[]').split(','))))
f.close()

#Deleting area younger then 90 yrs
AGE_DEL = 1
while AGE_DEL <=7:
    if post_burn_areas_DT6.has_key (AGE_DEL):
        del post_burn_areas_DT6[AGE_DEL]
    AGE_DEL=AGE_DEL+1

#Calculating old-growth area after burn
og_area_DT6=sum(post_burn_areas_DT6.values())
print "calculating HXS2 old-growth areas...."
print og_area_DT6
HXSOG_sum=HXSOG_sum + og_area_DT6
print "Total HXS old-growth so far..."
print HXSOG_sum

##update of next six development types
f=open("C:\\MC_CHP3_model\\ageclassp1c.txt","r")
prd = []; D7 = []; D8 = []; D9 = []; D10 = []; D11 = []; D12 = [];
for line in f:
    prdp, DD7, DD8, DD9, DD10, DD11, DD12 = line.split()
    prd.append(int(prdp))
    D7.append(float(DD7))
    D8.append(float(DD8))
    D9.append(float(DD9))
    D10.append(float(DD10))
    D11.append(float(DD11))
    D12.append(float(DD12))

f.close()

DT7=dict(zip(prd, D7))
DT8=dict(zip(prd, D8))
DT9=dict(zip(prd, D9))
DT10=dict(zip(prd, D10))
DT11=dict(zip(prd, D11))
DT12=dict(zip(prd, D12))
```

```
##*****************DT7**********************
##calculating burn area DT7
burn_areas_DT7 = dict(zip(DT7.keys(), \
        ([adjusted_fire*value for value in DT7.values()])))
print "Calculating burn areas for DT7...."
#print burn_areas_DT7

#summing burn area DT7
total_burn_area_DT7 = sum(burn_areas_DT7.values())
print "Summing burn areas..."
#print total_burn_area_DT7

#calculating new areas DT7
post_burn_areas_DT7 = dict(zip(DT7.keys(), \
        ([(value_1-(adjusted_fire*value_1)) for value_1 in DT7.values()])))
print "Calculating new post-burn areas..."
#print post_burn_areas

# Adding burn areas to age-class one DT7
post_burn_areas_DT7[1] = post_burn_areas_DT7[1]+ total_burn_area_DT7
print "Adding burn areas to age-class one...."
#print post_burn_areas_DT7

zero =0.0
look = Lookup(post_burn_areas_DT7)
deletlers = look.get_key(zero)
print deletlers

while deletlers:
    del post_burn_areas_DT7[deletlers.pop()]


#Generating new .are file post-burn
f=open("c:\\MC_CHP3_model\\OG_1.are","a")
for key in sorted(post_burn_areas_DT7):
    f.write('*A XSH1 ''%s %s\n' % (str(key), \
            ''.join(repr(post_burn_areas_DT7[key]).strip('[]').split(','))))
f.close()

#Deleting area younger then 90 yrs
AGE_DEL = 1
while AGE_DEL <=9:
    if post_burn_areas_DT7.has_key (AGE_DEL):
        del post_burn_areas_DT7[AGE_DEL]
    AGE_DEL=AGE_DEL+1

#Calculating old-growth area after burn
og_area_DT7=sum(post_burn_areas_DT7.values())
print "calculating XSH1 old-growth areas...."
print og_area_DT7
XSHOG_sum=og_area_DT7
print "Total XSH old-growth so far..."
print XSHOG_sum

##*****************DT8**********************
##calculating burn area DT8
burn_areas_DT8 = dict(zip(DT8.keys(), \
          ([adjusted_fire*value for value in DT8.values()])))
print "Calculating burn areas for DT8...."
#print burn_areas_DT8

#summing burn area DT8
total_burn_area_DT8 = sum(burn_areas_DT8.values())
print "Summing burn areas..."
#print total_burn_area_DT8

#calculating new areas DT8
post_burn_areas_DT8 = dict(zip(DT8.keys(), \
```

```
        ([(value_1-(adjusted_fire*value_1)) for value_1 in DT8.values()])))
print "Calculating new post-burn areas..."
#print post_burn_areas

# Adding burn areas to age-class one DT8
post_burn_areas_DT8[1] = post_burn_areas_DT8[1]+ total_burn_area_DT8
print "Adding burn areas to age-class one...."
#print post_burn_areas_DT8

zero =0.0
look = Lookup(post_burn_areas_DT8)
deletlers = look.get_key(zero)
print deletlers

while deletlers:
    del post_burn_areas_DT8[deletlers.pop()]


#Generating new .are file post-burn
f=open("c:\\MC_CHP3_model\\OG_1.are","a")
for key in sorted(post_burn_areas_DT8):
    f.write('*A XSH2 ''%s %s\n' % (str(key), \
            ''.join(repr(post_burn_areas_DT8[key]).strip('[]').split(','))))
f.close()

#Deleting area younger then 90 yrs
AGE_DEL = 1
while AGE_DEL <=9:
    if post_burn_areas_DT8.has_key (AGE_DEL):
        del post_burn_areas_DT8[AGE_DEL]
    AGE_DEL=AGE_DEL+1

#Calculating old-growth area after burn
og_area_DT8=sum(post_burn_areas_DT8.values())
print "calculating XSH2 old-growth areas...."
print og_area_DT8
XSHOG_sum=XSHOG_sum + og_area_DT8
print "Total XSH old-growth so far..."
print XSHOG_sum

##*****************DT9**********************
##calculating burn area DT9
burn_areas_DT9 = dict(zip(DT9.keys(), \
        ([adjusted_fire*value for value in DT9.values()])))
print "Calculating burn areas for DT9...."
#print burn_areas_DT9

#summing burn area DT9
total_burn_area_DT9 = sum(burn_areas_DT9.values())
print "Summing burn areas..."
#print total_burn_area_DT9

#calculating new areas DT9
post_burn_areas_DT9 = dict(zip(DT9.keys(), \
        ([(value_1-(adjusted_fire*value_1)) for value_1 in DT9.values()])))
print "Calculating new post-burn areas..."
#print post_burn_areas

# Adding burn areas to age-class one DT9
post_burn_areas_DT9[1] = post_burn_areas_DT9[1]+ total_burn_area_DT9
print "Adding burn areas to age-class one...."
#print post_burn_areas_DT9

zero =0.0
look = Lookup(post_burn_areas_DT9)
deletlers = look.get_key(zero)
print deletlers
```

```
while deletlers:
    del post_burn_areas_DT9[deletlers.pop()]


#Generating new .are file post-burn
f=open("c:\\MC_CHP3_model\\OG_1.are","a")
for key in sorted(post_burn_areas_DT9):
    f.write('*A BS1 ''%s %s\n' % (str(key), \
              ''.join(repr(post_burn_areas_DT9[key]).strip('[]').split(','))))
f.close()

#Deleting area younger then 90 yrs
AGE_DEL = 1
while AGE_DEL <=10:
    if post_burn_areas_DT9.has_key (AGE_DEL):
        del post_burn_areas_DT9[AGE_DEL]
    AGE_DEL=AGE_DEL+1

#Calculating old-growth area after burn
og_area_DT9=sum(post_burn_areas_DT9.values())
print "calculating BS1 old-growth areas...."
print og_area_DT9
BSOG_sum=og_area_DT9
print "Total BS old-growth so far..."
print BSOG_sum

##*****************DT10**********************
##calculating burn area DT10
burn_areas_DT10 = dict(zip(DT10.keys(), \
        ([adjusted_fire*value for value in DT10.values()])))
print "Calculating burn areas for DT10...."
#print burn_areas_DT10

#summing burn area DT10
total_burn_area_DT10 = sum(burn_areas_DT10.values())
print "Summing burn areas..."
#print total_burn_area_DT10

#calculating new areas DT10
post_burn_areas_DT10 = dict(zip(DT10.keys(), \
         ([(value_1-(adjusted_fire*value_1)) for value_1 in DT10.values()])))
print "Calculating new post-burn areas..."
#print post_burn_areas

# Adding burn areas to age-class one DT10
post_burn_areas_DT10[1] = post_burn_areas_DT10[1]+ total_burn_area_DT10
print "Adding burn areas to age-class one...."
#print post_burn_areas_DT10

zero =0.0
look = Lookup(post_burn_areas_DT10)
deletlers = look.get_key(zero)
print deletlers

while deletlers:
    del post_burn_areas_DT10[deletlers.pop()]


#Generating new .are file post-burn
f=open("c:\\MC_CHP3_model\\OG_1.are","a")
for key in sorted(post_burn_areas_DT10):
    f.write('*A BS2 ''%s %s\n' % (str(key), \
            ''.join(repr(post_burn_areas_DT10[key]).strip('[]').split(','))))
f.close()

#Deleting area younger then 90 yrs
AGE_DEL = 1
while AGE_DEL <=10:
```

```
    if post_burn_areas_DT10.has_key (AGE_DEL):
        del post_burn_areas_DT10[AGE_DEL]
    AGE_DEL=AGE_DEL+1

#Calculating old-growth area after burn
og_area_DT10=sum(post_burn_areas_DT10.values())
print "calculating BS2 old-growth areas...."
print og_area_DT10
BSOG_sum=BSOG_sum + og_area_DT10
print "Total BS old-growth so far..."
print BSOG_sum

##*****************DT11**********************
##calculating burn area DT11
burn_areas_DT11 = dict(zip(DT11.keys(), \
        ([adjusted_fire*value for value in DT11.values()])))
print "Calculating burn areas for DT11...."
#print burn_areas_DT11

#summing burn area DT11
total_burn_area_DT11 = sum(burn_areas_DT11.values())
print "Summing burn areas..."
#print total_burn_area_DT11

#calculating new areas DT11
post_burn_areas_DT11 = dict(zip(DT11.keys(), \
         ([(value_1-(adjusted_fire*value_1)) for value_1 in DT11.values()])))
print "Calculating new post-burn areas..."
#print post_burn_areas

# Adding burn areas to age-class one DT11
post_burn_areas_DT11[1] = post_burn_areas_DT11[1]+ total_burn_area_DT11
print "Adding burn areas to age-class one...."
#print post_burn_areas_DT11

zero =0.0
look = Lookup(post_burn_areas_DT11)
deletlers = look.get_key(zero)
print deletlers

while deletlers:
    del post_burn_areas_DT11[deletlers.pop()]


#Generating new .are file post-burn
f=open("c:\\MC_CHP3_model\\OG_1.are","a")
for key in sorted(post_burn_areas_DT11):
    f.write('*A JP1 ''%s %s\n' % (str(key), \
             ''.join(repr(post_burn_areas_DT11[key]).strip('[]').split(','))))
f.close()

#Deleting area younger then 90 yrs
AGE_DEL = 1
while AGE_DEL <=7:
    if post_burn_areas_DT11.has_key (AGE_DEL):
        del post_burn_areas_DT11[AGE_DEL]
    AGE_DEL=AGE_DEL+1

#Calculating old-growth area after burn
og_area_DT11=sum(post_burn_areas_DT11.values())
print "calculating JP1 old-growth areas...."
print og_area_DT11
JPOG_sum=og_area_DT11
print "Total JP old-growth so far..."
print JPOG_sum


##*****************DT12**********************
```

```
##calculating burn area DT12
burn_areas_DT12 = dict(zip(DT12.keys(), \
          ([adjusted_fire*value for value in DT12.values()])))
print "Calculating burn areas for DT12...."
#print burn_areas_DT12

#summing burn area DT12
total_burn_area_DT12 = sum(burn_areas_DT12.values())
print "Summing burn areas..."
#print total_burn_area_DT12

#calculating new areas DT12
post_burn_areas_DT12 = dict(zip(DT12.keys(), \
       ([(value_1-(adjusted_fire*value_1)) for value_1 in DT12.values()])))
print "Calculating new post-burn areas..."
#print post_burn_areas

# Adding burn areas to age-class one DT12
post_burn_areas_DT12[1] = post_burn_areas_DT12[1]+ total_burn_area_DT12
print "Adding burn areas to age-class one...."
#print post_burn_areas_DT12

zero =0.0
look = Lookup(post_burn_areas_DT12)
deletlers = look.get_key(zero)
print deletlers

while deletlers:
    del post_burn_areas_DT12[deletlers.pop()]

#Generating new .are file post-burn
f=open("c:\\MC_CHP3_model\\OG_1.are","a")
for key in sorted(post_burn_areas_DT12):
    f.write('*A JP2 ''%s %s\n' % (str(key), \
            ''.join(repr(post_burn_areas_DT12[key]).strip('[]').split(','))))
f.close()

#Deleting area younger then 90 yrs
AGE_DEL = 1
while AGE_DEL <=7:
    if post_burn_areas_DT12.has_key (AGE_DEL):
        del post_burn_areas_DT12[AGE_DEL]
    AGE_DEL=AGE_DEL+1

#Calculating old-growth area after burn
og_area_DT12=sum(post_burn_areas_DT12.values())
print "calculating JP2 old-growth areas...."
print og_area_DT12
JPOG_sum=JPOG_sum + og_area_DT12
print "Total JP old-growth so far..."
print JPOG_sum

##updating the last five development types
f=open("C:\\MC_CHP3_model\\ageclassp1b.txt","r")
prd = []; D13 = []; D14 = []; D15 = []; D16 = []; D17 = []; D18 = [];
for line in f:
    prdp, DD13, DD14, DD15, DD16, DD17, DD18 = line.split()
    prd.append(int(prdp))
    D13.append(float(DD13))
    D14.append(float(DD14))
    D15.append(float(DD15))
    D16.append(float(DD16))
    D17.append(float(DD17))
    D18.append(float(DD18))
f.close()

DT13=dict(zip(prd, D13))
DT14=dict(zip(prd, D14))
```

```
DT15=dict(zip(prd, D15))
DT16=dict(zip(prd, D16))
DT17=dict(zip(prd, D17))
DT18=dict(zip(prd, D18))

##*****************DT13**********************
##calculating burn area DT13
burn_areas_DT13 = dict(zip(DT13.keys(), \
         ([adjusted_fire*value for value in DT13.values()])))
print "Calculating burn areas for DT13...."
#print burn_areas_DT13

#summing burn area DT13
total_burn_area_DT13 = sum(burn_areas_DT13.values())
print "Summing burn areas..."
#print total_burn_area_DT13

#calculating new areas DT13
post_burn_areas_DT13 = dict(zip(DT13.keys(), \
         ([(value_1-(adjusted_fire*value_1)) for value_1 in DT13.values()])))
print "Calculating new post-burn areas..."
#print post_burn_areas

# Adding burn areas to age-class one DT13
post_burn_areas_DT13[1] = post_burn_areas_DT13[1]+ total_burn_area_DT13
print "Adding burn areas to age-class one...."
#print post_burn_areas_DT13

zero =0.0
look = Lookup(post_burn_areas_DT13)
deletlers = look.get_key(zero)
print deletlers

while deletlers:
    del post_burn_areas_DT13[deletlers.pop()]


#Generating new .are file post-burn
f=open("c:\\MC_CHP3_model\\OG_1.are","a")
for key in sorted(post_burn_areas_DT13):
    f.write('*A JPBS1 ''%s %s\n' % (str(key), \
             ''.join(repr(post_burn_areas_DT13[key]).strip('[]').split(','))))
f.close()

#Deleting area younger then 90 yrs
AGE_DEL = 1
while AGE_DEL <=10:
    if post_burn_areas_DT13.has_key (AGE_DEL):
        del post_burn_areas_DT13[AGE_DEL]
    AGE_DEL=AGE_DEL+1

#Calculating old-growth area after burn
og_area_DT13=sum(post_burn_areas_DT13.values())
print "calculating JPBS1 old-growth areas...."
print og_area_DT13
JPBSOG_sum=og_area_DT13
print "Total JPBS old-growth so far..."
print JPBSOG_sum

##*****************DT14**********************
##calculating burn area DT14
burn_areas_DT14 = dict(zip(DT14.keys(), \
         ([adjusted_fire*value for value in DT14.values()])))
print "Calculating burn areas for DT14...."
#print burn_areas_DT14

#summing burn area DT14
total_burn_area_DT14 = sum(burn_areas_DT14.values())
```

```
print "Summing burn areas..."
#print total_burn_area_DT14

#calculating new areas DT14
post_burn_areas_DT14 = dict(zip(DT14.keys(), \
        ([(value_1-(adjusted_fire*value_1)) for value_1 in DT14.values()])))
print "Calculating new post-burn areas..."
#print post_burn_areas

# Adding burn areas to age-class one DT14
post_burn_areas_DT14[1] = post_burn_areas_DT14[1]+ total_burn_area_DT14
print "Adding burn areas to age-class one...."
#print post_burn_areas_DT14

zero =0.0
look = Lookup(post_burn_areas_DT14)
deletlers = look.get_key(zero)
print deletlers

while deletlers:
    del post_burn_areas_DT14[deletlers.pop()]


#Generating new .are file post-burn
f=open("c:\\MC_CHP3_model\\OG_1.are","a")
for key in sorted(post_burn_areas_DT14):
    f.write('*A JPBS2 ''%s %s\n' % (str(key), \
             ''.join(repr(post_burn_areas_DT14[key]).strip('[]').split(','))))
f.close()

#Deleting area younger then 90 yrs
AGE_DEL = 1
while AGE_DEL <=10:
    if post_burn_areas_DT14.has_key (AGE_DEL):
        del post_burn_areas_DT14[AGE_DEL]
    AGE_DEL=AGE_DEL+1

#Calculating old-growth area after burn
og_area_DT14=sum(post_burn_areas_DT14.values())
print "calculating JPBS2 old-growth areas...."
print og_area_DT14
JPBSOG_sum=JPBSOG_sum + og_area_DT14
print "Total JPBS old-growth so far..."
print JPBSOG_sum

##*****************DT15**********************
##calculating burn area DT15
burn_areas_DT15 = dict(zip(DT15.keys(), \
         ([adjusted_fire*value for value in DT15.values()])))
print "Calculating burn areas for DT15...."
#print burn_areas_DT15

#summing burn area DT15
total_burn_area_DT15 = sum(burn_areas_DT15.values())
print "Summing burn areas..."
#print total_burn_area_DT15

#calculating new areas DT15
post_burn_areas_DT15 = dict(zip(DT15.keys(), \
         ([(value_1-(adjusted_fire*value_1)) for value_1 in DT15.values()])))
print "Calculating new post-burn areas..."
#print post_burn_areas

# Adding burn areas to age-class one DT15
post_burn_areas_DT15[1] = post_burn_areas_DT15[1]+ total_burn_area_DT15
print "Adding burn areas to age-class one...."
#print post_burn_areas_DT15
```

```
zero =0.0
look = Lookup(post_burn_areas_DT15)
deletlers = look.get_key(zero)
print deletlers

while deletlers:
    del post_burn_areas_DT15[deletlers.pop()]


#Generating new .are file post-burn
f=open("c:\\MC_CHP3_model\\OG_1.are","a")
for key in sorted(post_burn_areas_DT15):
    f.write('*A WSBF1 ''%s %s\n' % (str(key), \
              ''.join(repr(post_burn_areas_DT15[key]).strip('[]').split(','))))
f.close()

#Deleting area younger then 90 yrs
AGE_DEL = 1
while AGE_DEL <=9:
    if post_burn_areas_DT15.has_key (AGE_DEL):
        del post_burn_areas_DT15[AGE_DEL]
    AGE_DEL=AGE_DEL+1

#Calculating old-growth area after burn
og_area_DT15=sum(post_burn_areas_DT15.values())
print "calculating WSBF1 old-growth areas...."
print og_area_DT15
WSBFOG_sum=og_area_DT15
print "Total WSBF old-growth so far..."
print WSBFOG_sum

##*****************DT16**********************
##calculating burn area DT16
burn_areas_DT16 = dict(zip(DT16.keys(), \
         ([adjusted_fire*value for value in DT16.values()])))
print "Calculating burn areas for DT16...."
#print burn_areas_DT16

#summing burn area DT16
total_burn_area_DT16 = sum(burn_areas_DT16.values())
print "Summing burn areas..."
#print total_burn_area_DT16

#calculating new areas DT16
post_burn_areas_DT16 = dict(zip(DT16.keys(), \
       ([(value_1-(adjusted_fire*value_1)) for value_1 in DT16.values()])))
print "Calculating new post-burn areas..."
#print post_burn_areas

# Adding burn areas to age-class one DT16
post_burn_areas_DT16[1] = post_burn_areas_DT16[1]+ total_burn_area_DT16
print "Adding burn areas to age-class one...."
#print post_burn_areas_DT16

zero =0.0
look = Lookup(post_burn_areas_DT16)
deletlers = look.get_key(zero)
print deletlers

while deletlers:
    del post_burn_areas_DT16[deletlers.pop()]


#Generating new .are file post-burn
f=open("c:\\MC_CHP3_model\\OG_1.are","a")
for key in sorted(post_burn_areas_DT16):
    f.write('*A WSBF2 ''%s %s\n' % (str(key), \
             ''.join(repr(post_burn_areas_DT16[key]).strip('[]').split(','))))
```

```
f.close()

#Deleting area younger then 90 yrs
AGE_DEL = 1
while AGE_DEL <=9:
    if post_burn_areas_DT16.has_key (AGE_DEL):
        del post_burn_areas_DT16[AGE_DEL]
    AGE_DEL=AGE_DEL+1

#Calculating old-growth area after burn
og_area_DT16=sum(post_burn_areas_DT16.values())
print "calculating WSBF2 old-growth areas...."
print og_area_DT16
WSBFOG_sum=WSBFOG_sum + og_area_DT16
print "Total WSBF old-growth so far..."
print WSBFOG_sum

##*****************DT17**********************
##calculating burn area DT17
burn_areas_DT17 = dict(zip(DT17.keys(), \
          ([adjusted_fire*value for value in DT17.values()])))
print "Calculating burn areas for DT17...."
#print burn_areas_DT17

#summing burn area DT17
total_burn_area_DT17 = sum(burn_areas_DT17.values())
print "Summing burn areas..."
#print total_burn_area_DT17

#calculating new areas DT17
post_burn_areas_DT17 = dict(zip(DT17.keys(), \
         ([(value_1-(adjusted_fire*value_1)) for value_1 in DT17.values()])))
print "Calculating new post-burn areas..."
#print post_burn_areas

# Adding burn areas to age-class one DT17
post_burn_areas_DT17[1] = post_burn_areas_DT17[1]+ total_burn_area_DT17
print "Adding burn areas to age-class one...."
#print post_burn_areas_DT17

zero =0.0
look = Lookup(post_burn_areas_DT17)
deletlers = look.get_key(zero)
print deletlers

while deletlers:
    del post_burn_areas_DT17[deletlers.pop()]


#Generating new .are file post-burn
f=open("c:\\MC_CHP3_model\\OG_1.are","a")
for key in sorted(post_burn_areas_DT17):
    f.write('*A TL1 ''%s %s\n' % (str(key), \
             ''.join(repr(post_burn_areas_DT17[key]).strip('[]').split(','))))
f.close()

#Deleting area younger then 90 yrs
AGE_DEL = 1
while AGE_DEL <=10:
    if post_burn_areas_DT17.has_key (AGE_DEL):
        del post_burn_areas_DT17[AGE_DEL]
    AGE_DEL=AGE_DEL+1

#Calculating old-growth area after burn
og_area_DT17=sum(post_burn_areas_DT17.values())
print "calculating TL1 old-growth areas...."
print og_area_DT17
TLOG_sum=og_area_DT17
```

```
    print "Total TL old-growth so far..."
    print TLOG_sum

    ##*****************DT18**********************
    ##calculating burn area DT18
    burn_areas_DT18 = dict(zip(DT18.keys(), \
              ([adjusted_fire*value for value in DT18.values()])))
    print "Calculating burn areas for DT18...."
    #print burn_areas_DT18

    #summing burn area DT18
    total_burn_area_DT18 = sum(burn_areas_DT18.values())
    print "Summing burn areas..."
    #print total_burn_area_DT18

    #calculating new areas DT18
    post_burn_areas_DT18 = dict(zip(DT18.keys(), \
             ([(value_1-(adjusted_fire*value_1)) for value_1 in DT18.values()])))
    print "Calculating new post-burn areas..."
    #print post_burn_areas

    # Adding burn areas to age-class one DT18
    post_burn_areas_DT18[1] = post_burn_areas_DT18[1]+ total_burn_area_DT18
    print "Adding burn areas to age-class one...."
    #print post_burn_areas_DT18

    zero =0.0
    look = Lookup(post_burn_areas_DT18)
    deletlers = look.get_key(zero)
    print deletlers

    while deletlers:
        del post_burn_areas_DT18[deletlers.pop()]


    #Generating new .are file post-burn
    f=open("c:\\MC_CHP3_model\\OG_1.are","a")
    for key in sorted(post_burn_areas_DT18):
        f.write('*A TL2 ''%s %s\n' % (str(key), \
                ''.join(repr(post_burn_areas_DT18[key]).strip('[]').split(','))))
    f.close()

    #Deleting area younger then 90 yrs
    AGE_DEL = 1
    while AGE_DEL <=10:
        if post_burn_areas_DT18.has_key (AGE_DEL):
            del post_burn_areas_DT18[AGE_DEL]
        AGE_DEL=AGE_DEL+1

    #Calculating old-growth area after burn
    og_area_DT18=sum(post_burn_areas_DT18.values())
    print "calculating TL2 old-growth areas...."
    print og_area_DT18
    TLOG_sum=TLOG_sum + og_area_DT18
    print "Total S old-growth so far..."
    print TLOG_sum


#***************writing results file****************************************

    #writing the results file - [draw_#] [planning_period] [OG_area] [harvest volume]
    og=open(r'c:\MC_CHP3_model\OG_results.txt','a')
    str_pp=str(planning_period)
    str_draw=str(draw)
    str_HWog_area=str(HWOG_sum)
    str_HJPog_area=str(HJPOG_sum)
    str_JPHog_area=str(JPHOG_sum)
    str_HXSog_area=str(HXSOG_sum)
```

```
    str_XSHog_area=str(XSHOG_sum)
    str_BSog_area=str(BSOG_sum)
    str_JPog_area=str(JPOG_sum)
    str_JPBSog_area=str(JPBSOG_sum)
    str_WSBFog_area=str(WSBFOG_sum)
    str_TLog_area=str(TLOG_sum)

    str_harv_totsplt=str(harv_totsplt_flt)
    str_rev_totsplt=str(rev_totsplt_flt)
    str_PVrev_totsplt=str(PVrev_totsplt_flt)
    str_cost_totsplt=str(cost_totsplt_flt)
    str_PVcost_totsplt=str(PVcost_totsplt_flt)
    str_NPV_totsplt=str(NPV_totsplt_flt)
    str_total_fire=str(total_fire)
    og.write(str_draw)
    og.write(" ")
    og.write(str_pp)
    og.write(" ")
    og.write(str_HWog_area)
    og.write(" ")
    og.write(str_HJPog_area)
    og.write(" ")
    og.write(str_JPHog_area)
    og.write(" ")
    og.write(str_HXSog_area)
    og.write(" ")
    og.write(str_XSHog_area)
    og.write(" ")
    og.write(str_BSog_area)
    og.write(" ")
    og.write(str_JPog_area)
    og.write(" ")
    og.write(str_JPBSog_area)
    og.write(" ")
    og.write(str_WSBFog_area)
    og.write(" ")
    og.write(str_TLog_area)
    og.write(" ")
    og.write(str_harv_totsplt)
    og.write(" ")
    og.write(str_rev_totsplt)
    og.write(" ")
    og.write(str_PVrev_totsplt)
    og.write(" ")
    og.write(str_cost_totsplt)
    og.write(" ")
    og.write(str_PVcost_totsplt)
    og.write(" ")
    og.write(str_NPV_totsplt)
    og.write(" ")
    og.write(str_total_fire)
    og.write("\n")
    og.close()

    planning_period = planning_period + 1

draw = draw+1
```

# Appendix F

# Chapter 4 Woodstock Model Formulation

```
ACTIONS

*ACTION harvest Y Harvest
  *OPERABLE harvest
  HW1 TOt >= 60
  HW2 TOt >= 60
  HJP TOt >= 60
  JPH TOt >= 60
  HXS1 TOt >= 60
  HXS2 TOt >= 60
  XSH1 TOt >= 60
  XSH2 TOt >= 60
  BS1 TOt >= 60
  BS2 TOt >= 60
  JP1 TOt >= 60
  JP2 TOt >= 60
  JPBS1 TOt >= 60
  JPBS2 TOt >= 60
  WSBF1 TOt >= 60
  WSBF2 TOt >= 60
  TL1 TOt >= 60
  TL2 TOt >= 60

*ACTION burn n
 *OPERABLE burn
  ? _CP >= 1

AREAS
*A HW1 1 26787.0207
*A HW1 2 22721.5768
*A HW1 3 10676.6962
*A HW1 4 15416.3405
*A HW1 5 5932.6899
*A HW1 6 10007.3756
*A HW1 7 29476.9515
*A HW1 8 38431.4844
*A HW1 9 20534.8266
*A HW1 10 8323.4428
*A HW1 11 10612.8143
*A HW1 12 5476.0103
*A HW1 13 4484.4766
*A HW2 1 23170.8924
*A HW2 2 20154.1853
*A HW2 3 7539.9891
*A HW2 4 4757.1029
```

*A HW2 5 4978.1103
*A HW2 6 10026.4539
*A HW2 7 23771.7336
*A HW2 8 23606.2036
*A HW2 9 13813.3597
*A HW2 10 8087.3908
*A HW2 11 7444.4501
*A HW2 12 12125.0968
*A HW2 13 6555.5613
*A HJP 1 15632.0887
*A HJP 2 11874.07
*A HJP 3 6721.5131
*A HJP 4 3401.7001
*A HJP 5 450.6933
*A HJP 6 505.0641
*A HJP 7 4031.7085
*A HJP 8 4292.2322
*A HJP 9 1770.9444
*A HJP 10 746.3531
*A HJP 11 1032.9663
*A HJP 12 927.017
*A HJP 13 367.6934
*A HJP 14 152.6594
*A JPH 1 9399.0196
*A JPH 2 9978.4356
*A JPH 3 6162.7919
*A JPH 4 3313.9969
*A JPH 5 444.5051
*A JPH 6 363.2827
*A JPH 7 2990.69
*A JPH 8 2852.7594
*A JPH 9 1044.0975
*A JPH 10 658.7505
*A JPH 11 816.5678
*A JPH 12 1358.3244
*A JPH 13 710.1209
*A JPH 14 276.6655
*A HXS1 1 3170.071
*A HXS1 2 1459.875
*A HXS1 3 904.3762
*A HXS1 4 1328.9817
*A HXS1 5 541.0714
*A HXS1 6 174.6266
*A HXS1 7 2872.8753
*A HXS1 8 2450.6623
*A HXS1 9 1846.8255
*A HXS1 10 1711.3275
*A HXS1 11 4618.6651
*A HXS1 12 4938.215
*A HXS1 13 3257.4915
*A HXS1 14 1075.8297
*A HXS2 1 10473.3923
*A HXS2 2 2818.6194
*A HXS2 3 915.6045
*A HXS2 4 2142.945
*A HXS2 5 1232.7402
*A HXS2 6 2871.6311
*A HXS2 7 6570.7086
*A HXS2 8 7476.0266
*A HXS2 9 5040.4053
*A HXS2 10 3352.4922
*A HXS2 11 5929.4184
*A HXS2 12 9738.5676
*A HXS2 13 3495.3755
*A HXS2 14 1003.2359
*A XSH1 1 1381.8142
*A XSH1 2 332.4182
*A XSH1 3 172.6488

```
*A XSH1 4 334.5231
*A XSH1 5 134.1423
*A XSH1 6 41.5465
*A XSH1 7 1374.6144
*A XSH1 8 1037.7396
*A XSH1 9 622.146
*A XSH1 10 881.0643
*A XSH1 11 1514.9898
*A XSH1 12 2027.6211
*A XSH1 13 2206.1017
*A XSH1 14 879.7117
*A XSH1 15 466.9994
*A XSH2 1 5148.2988
*A XSH2 2 913.0708
*A XSH2 3 251.1983
*A XSH2 4 424.223
*A XSH2 5 497.2657
*A XSH2 6 1043.6843
*A XSH2 7 3026.9726
*A XSH2 8 3305.4379
*A XSH2 9 2128.6083
*A XSH2 10 1657.1119
*A XSH2 11 2600.5282
*A XSH2 12 6385.4427
*A XSH2 13 3968.2855
*A XSH2 14 1435.1247
*A XSH2 15 806.8819
*A BS1 1 6399.1656
*A BS1 2 577.9363
*A BS1 3 1132.6113
*A BS1 4 560.0627
*A BS1 5 345.4056
*A BS1 6 976.1463
*A BS1 7 9030.6112
*A BS1 8 11799.5024
*A BS1 9 10735.0143
*A BS1 10 9233.568
*A BS1 11 19598.9339
*A BS1 12 24070.0914
*A BS1 13 11421.269
*A BS1 14 5183.5126
*A BS1 15 2582.2666
*A BS1 16 478.5025
*A BS1 17 295.6675
*A BS1 18 269.8559
*A BS2 1 15500.578
*A BS2 2 3339.4769
*A BS2 3 1489.885
*A BS2 4 566.9609
*A BS2 5 529.1735
*A BS2 6 1733.5786
*A BS2 7 10707.7841
*A BS2 8 10785.8607
*A BS2 9 5276.569
*A BS2 10 5312.3925
*A BS2 11 10724.135
*A BS2 12 12401.8822
*A BS2 13 5180.5826
*A BS2 14 2388.4889
*A BS2 15 672.1129
*A BS2 16 116.8134
*A BS2 17 24.7271
*A BS2 18 55.7583
*A JP1 1 1664.7486
*A JP1 2 941.6979
*A JP1 3 10835.083
*A JP1 4 15428.1018
*A JP1 5 6498.1406
```

*A JP1 6 566.2657
*A JP1 7 5621.1492
*A JP1 8 4730.3035
*A JP1 9 1574.8384
*A JP1 10 1732.3382
*A JP1 11 1451.1895
*A JP1 12 1011.1508
*A JP1 13 622.1015
*A JP1 14 213.507
*A JP1 15 177.3335
*A JP2 1 54000.7368
*A JP2 2 35094.6012
*A JP2 3 23273.5906
*A JP2 4 6367.1073
*A JP2 5 3286.5128
*A JP2 6 3713.5749
*A JP2 7 48458.9025
*A JP2 8 33306.8077
*A JP2 9 15878.1081
*A JP2 10 8765.7526
*A JP2 11 5922.8651
*A JP2 12 8165.9164
*A JP2 13 5476.0661
*A JP2 14 2043.0938
*A JP2 15 559.9986
*A JPBS1 1 4366.6704
*A JPBS1 2 3084.4122
*A JPBS1 3 3680.7391
*A JPBS1 4 1974.6499
*A JPBS1 5 286.4743
*A JPBS1 6 248.8239
*A JPBS1 7 2356.0979
*A JPBS1 8 1874.6592
*A JPBS1 9 1373.4279
*A JPBS1 10 863.4857
*A JPBS1 11 1654.761
*A JPBS1 12 2817.96
*A JPBS1 13 2200.0042
*A JPBS1 14 1103.8837
*A JPBS1 15 608.1172
*A JPBS1 16 236.3578
*A JPBS2 1 21199.812
*A JPBS2 2 9024.0228
*A JPBS2 3 3290.1786
*A JPBS2 4 3420.0505
*A JPBS2 5 4310.3174
*A JPBS2 6 3228.7197
*A JPBS2 7 46149.3696
*A JPBS2 8 31947.2643
*A JPBS2 9 10639.6601
*A JPBS2 10 6670.7579
*A JPBS2 11 9790.1302
*A JPBS2 12 12175.8386
*A JPBS2 13 6587.2998
*A JPBS2 14 2399.9043
*A JPBS2 15 902.3302
*A JPBS2 16 200.5518
*A WSBF1 1 1677.3096
*A WSBF1 2 118.663
*A WSBF1 3 45.6636
*A WSBF1 4 106.6825
*A WSBF1 5 55.6591
*A WSBF1 6 38.9486
*A WSBF1 7 682.0565
*A WSBF1 8 585.6799
*A WSBF1 9 704.5503
*A WSBF1 10 995.172
*A WSBF1 11 1327.8681

```
*A WSBF1 12 2492.5319
*A WSBF1 13 3079.266
*A WSBF1 14 1686.7296
*A WSBF1 15 602.1475
*A WSBF1 16 368.982
*A WSBF1 17 287.9655
*A WSBF2 1 9436.8222
*A WSBF2 2 589.9183
*A WSBF2 3 116.2899
*A WSBF2 4 510.5
*A WSBF2 5 622.8171
*A WSBF2 6 827.023
*A WSBF2 7 1211.4108
*A WSBF2 8 2455.5562
*A WSBF2 9 2090.9813
*A WSBF2 10 1825.5597
*A WSBF2 11 3298.6419
*A WSBF2 12 6014.4722
*A WSBF2 13 4929.5486
*A WSBF2 14 1685.5491
*A WSBF2 15 1144.8839
*A WSBF2 16 119.8256
*A WSBF2 17 68.5231
*A TL1 1 290.2901
*A TL1 2 173.1491
*A TL1 3 1231.4023
*A TL1 4 371.8533
*A TL1 5 147.6303
*A TL1 6 468.3224
*A TL1 7 783.6679
*A TL1 8 838.93
*A TL1 9 1495.7606
*A TL1 10 600.2179
*A TL1 11 2009.4409
*A TL1 12 2532.8975
*A TL1 13 1802.0845
*A TL1 14 949.3074
*A TL1 15 735.1381
*A TL2 1 975.9905
*A TL2 2 1163.2922
*A TL2 3 733.785
*A TL2 4 263.6752
*A TL2 5 260.6255
*A TL2 6 907.3635
*A TL2 7 4058.0356
*A TL2 8 3273.4304
*A TL2 9 5895.1031
*A TL2 10 3460.8267
*A TL2 11 7899.0042
*A TL2 12 8701.872
*A TL2 13 3168.4891
*A TL2 14 1154.7775
*A TL2 15 705.0129

CONTROL
; Control
 *LENGTH 20
 *GRAPHICS ON
 *REPORTS ON
 *IMAGE OFF
 *BUILD OFF
 *OPTIMIZE OFF
 *SCHEDULE ON
 *QUEUE OFF


LANDSCAPE
```

```
*THEME covtyp
  HW1
  HW2
  HJP
  JPH
  HXS1
  HXS2
  XSH1
  XSH2
  BS1
  BS2
  JP1
  JP2
  JPBS1
  JPBS2
  WSBF1
  WSBF2
  TL1
  TL2

*AGGREGATE HW
 HW1
 HW2

*AGGREGATE HXS
 HXS1
 HXS2

*AGGREGATE XSH
 XSH1
 XSH2

*AGGREGATE BS
 BS1
 BS2

*AGGREGATE JP
 JP1
 JP2

*AGGREGATE JPBS
 JPBS1
 JPBS2

*AGGREGATE WSBF
 WSBF1
 WSBF2

*AGGREGATE TL
 TL1
 TL2


LIFESPAN

HW1 13
HW2 13
HJP 14
JPH 14
HXS1 14
HXS2 14
XSH1 15
XSH2 15
BS1 18
BS2 18
JP1 15
JP2 15
JPBS1 16
```

```
JPBS2 16
WSBF1 17
WSBF2 17
TL1 15
TL2 15


OPTIMIZE

 *OBJECTIVE
 _MAX oPVTotalRev - oPVTotalCost  1.._LENGTH

 *CONSTRAINTS
;****Old forest constraints****
 HWoldgrowtharea >= 22495 1.._LENGTH
 HJPoldgrowtharea >= 3114 1.._LENGTH
 JPHoldgrowtharea >= 2422 1.._LENGTH
 HXSoldgrowtharea >= 5605 1.._LENGTH
 XSHoldgrowtharea >= 2820 1.._LENGTH
 BSoldgrowtharea >= 12090 1.._LENGTH
 JPoldgrowtharea >= 18443 1.._LENGTH
 JPBSoldgrowtharea >= 12040 1.._LENGTH
 WSBFoldgrowtharea >= 6216 1.._LENGTH
 TLoldgrowtharea >= 3423 1.._LENGTH

;****Even-flow constraints****
 _EVEN(softvol) 1.._LENGTH
 _EVEN(hardvol) 1.._LENGTH

;****Non-declining growing stock constraint****
 _NDY(standinginv) 15.._LENGTH

;****Area burned constraint****
burnarea = Totalarea 1.._LENGTH

 *EXCLUDE
;burn 1.._LENGTH

*FORMAT MOSEK


OUTPUTS

*OUTPUT totvol total volume cut
 *SOURCE harvest tot

*OUTPUT softvol softwood volume cut
 *SOURCE harvest Ysoft

*OUTPUT hardvol hardwood volume cut
 *SOURCE harvest Yhard

*OUTPUT standinginv
 *SOURCE _INVENT tot

*OUTPUT burnarea
*SOURCE burn _AREA

*OUTPUT TotalArea
 *SOURCE _INVENT _AREA

*OUTPUT HWarea
 *SOURCE HW _INVENT _AREA

*OUTPUT HJParea
 *SOURCE HJP _INVENT _AREA

*OUTPUT HXSarea
```

```
 *SOURCE HXS _INVENT _AREA

*OUTPUT JPHarea
 *SOURCE JPH _INVENT _AREA

*OUTPUT XSHarea
 *SOURCE XSH _INVENT _AREA

*OUTPUT BSarea
 *SOURCE BS _INVENT _AREA

*OUTPUT JParea
 *SOURCE JP _INVENT _AREA

*OUTPUT JPBSarea
 *SOURCE JPBS _INVENT _AREA

*OUTPUT WSBFarea
 *SOURCE WSBF _INVENT _AREA

*OUTPUT TLarea
 *SOURCE TL _INVENT _AREA

;****Yield group old forest definitions****

*OUTPUT HWoldgrowtharea
 *SOURCE HW @AGE(8.._MAXAGE) _INVENT _AREA

*OUTPUT HJPoldgrowtharea
 *SOURCE HJP @AGE(8.._MAXAGE) _INVENT _AREA

*OUTPUT JPHoldgrowtharea
 *SOURCE JPH @AGE(8.._MAXAGE) _INVENT _AREA

*OUTPUT HXSoldgrowtharea
 *SOURCE HXS @AGE(8.._MAXAGE) _INVENT _AREA

*OUTPUT XSHoldgrowtharea
 *SOURCE XSH @AGE(10.._MAXAGE) _INVENT _AREA

*OUTPUT BSoldgrowtharea
 *SOURCE BS @AGE(11.._MAXAGE) _INVENT _AREA

*OUTPUT JPoldgrowtharea
 *SOURCE JP @AGE(8.._MAXAGE) _INVENT _AREA

*OUTPUT JPBSoldgrowtharea
 *SOURCE JPBS @AGE(11.._MAXAGE) _INVENT _AREA

*OUTPUT WSBFoldgrowtharea
 *SOURCE WSBF @AGE(10.._MAXAGE) _INVENT _AREA

*OUTPUT TLoldgrowtharea
 *SOURCE TL @AGE(11.._MAXAGE) _INVENT _AREA


;****Revenue****
*OUTPUT oTotalRevenue Total revenue
 *SOURCE ? harvest tot * ytot$

*OUTPUT oPVTotalRev Present value of total revenue
 *SOURCE oTotalRevenue * yDisc5%

;****Cost****

*OUTPUT osilv Cost Total cost of establisment (site prep, plant)
 *SOURCE ? harvest ysilv$
```

```
*OUTPUT oHarvCost Total cost of harvesting
 *SOURCE ? harvest yHarvcost$

*OUTPUT oTotalCost Grand total costs
 *SOURCE osilvCost + oHarvcost

*OUTPUT oPVTotalCost Present value of total costs
 *SOURCE oTotalCost * yDisc5%

*OUTPUT oNPV Net Present Value
 *SOURCE oPVTotalrev - oPVtotalcost


REPORTS

*TARGET OG_1_allrep.txt
_ALL 1.._LENGTH

*TARGET surplus.wk1
_ALL   1.._LENGTH

*TARGET ageclassP1.txt
 _AGECLASS HW1 1
 _AGECLASS HW2 1
 _AGECLASS HJP 1
 _AGECLASS JPH 1
 _AGECLASS HXS1 1
 _AGECLASS HXS2 1
 _AGECLASS XSH1 1
 _AGECLASS XSH2 1
 _AGECLASS BS1 1
 _AGECLASS BS2 1
 _AGECLASS JP1 1
 _AGECLASS JP2 1
 _AGECLASS JPBS1 1
 _AGECLASS JPBS2 1
 _AGECLASS WSBF1 1
 _AGECLASS WSBF2 1
 _AGECLASS TL1 1
 _AGECLASS TL2 1

*TARGET HWogreport.txt
 HWoldgrowtharea

*TARGET HJPogreport.txt
 HJPoldgrowtharea

*TARGET HXSogreport.txt
 HXSoldgrowtharea

*TARGET JPHogreport.txt
 JPHoldgrowtharea

*TARGET XSHogreport.txt
 XSHoldgrowtharea

*TARGET BSogreport.txt
 BSoldgrowtharea

*TARGET JPogreport.txt
 JPoldgrowtharea

*TARGET JPBSogreport.txt
 JPBSoldgrowtharea

*TARGET WSBFogreport.txt
 WSBFoldgrowtharea
```

```
*TARGET TLogreport.txt
 TLoldgrowtharea

*TARGET Harvreport.txt
 totvol 1

*TARGET revenuereport.txt
 ototalrevenue 1

*TARGET costreport.txt
 ototalcost 1

*TARGET PVrevenuereport.txt
 oPVtotalrev 1

*TARGET PVcostreport.txt
 oPVtotalcost 1

*TARGET NPV.txt
 oNPV 1

*TARGET SCHEDULE.txt
 _SCHEDULE 1


TRANSITIONS

*CASE _DEATH
 *SOURCE ?
 *TARGET ?  100

*CASE harvest
 *SOURCE ?
 *TARGET ? 100

*CASE BURN
*SOURCE ?
*TARGET ? 94
*TARGET ? 6 _AGE 0


YIELDS

*Y HW1 ; Set number 1
_AGE     YHARD       YSOFT       Tot
   1     5.81111     0.00000     5.81111
   2    24.41943     0.00000    24.41943
   3    48.12660     2.51409    50.64069
   4    71.16462     7.48369    78.64831
  5    89.93458    14.31431   104.24889
  6   102.90608    22.10001   125.00608
  7   109.94592    29.93436   139.88028
  8   111.71794    37.08057   148.79851
  9   109.24987    43.03283   152.28271
  10   101.43652    43.19294   144.62946
  11    90.75899    38.64631   129.40531
  12    80.08146    34.09969   114.18115
  13    69.40394    29.55306    98.95700
  14    58.72641    25.00644    83.73285
  15    48.04888    20.45981    68.50869
  16    37.37135    15.91319    53.28454
  17    26.69382    11.36656    38.06038
  18    16.01629     6.81994    22.83623
  19     5.33876     2.27331     7.61208

*Y HW2   ; Set number 2
_AGE YHARD  ysoft  Tot
   1     7.29466     0.00000     7.29466
```

| | | | |
|---|---|---|---|
| 2 | 31.17993 | 0.00000 | 31.17993 |
| 3 | 62.79053 | 1.73584 | 64.52637 |
| 4 | 93.05879 | 6.16787 | 99.22666 |
| 5 | 117.28297 | 12.39038 | 129.67335 |
| 6 | 133.41111 | 19.47693 | 152.88805 |
| 7 | 141.39235 | 26.50723 | 167.89958 |
| 8 | 142.28935 | 32.75489 | 175.04424 |
| 9 | 137.63970 | 37.74908 | 175.38878 |
| 10 | 127.05496 | 37.71190 | 164.76686 |
| 11 | 113.68075 | 33.74223 | 147.42298 |
| 12 | 100.30655 | 29.77255 | 130.07910 |
| 13 | 86.93234 | 25.80288 | 112.73522 |
| 14 | 73.55813 | 21.83321 | 95.39134 |
| 15 | 60.18393 | 17.86353 | 78.04746 |
| 16 | 46.80972 | 13.89386 | 60.70358 |
| 17 | 33.43552 | 9.92418 | 43.35970 |
| 18 | 20.06131 | 5.95451 | 26.01582 |
| 19 | 6.68710 | 1.98484 | 8.67194 |

*Y HJP ; Set number 3

| _AGE | YHARD | ysoft | Tot |
|---|---|---|---|
| 1 | 2.34764 | 1.36888 | 3.71652 |
| 2 | 9.16676 | 7.33538 | 16.50214 |
| 3 | 18.94295 | 17.76022 | 36.70317 |
| 4 | 30.16721 | 31.34230 | 61.50951 |
| 5 | 41.65106 | 46.62139 | 88.27245 |
| 6 | 52.54153 | 62.30136 | 114.84289 |
| 7 | 62.27561 | 77.35557 | 139.63118 |
| 8 | 70.52222 | 91.04056 | 161.56278 |
| 9 | 77.12691 | 102.86974 | 179.99665 |
| 10 | 75.81151 | 102.59663 | 178.40814 |
| 11 | 67.83136 | 91.79698 | 159.62834 |
| 12 | 59.85120 | 80.99734 | 140.84853 |
| 13 | 51.87104 | 70.19769 | 122.06873 |
| 14 | 43.89088 | 59.39805 | 103.28893 |
| 15 | 35.91072 | 48.59840 | 84.50912 |
| 16 | 27.93056 | 37.79876 | 65.72932 |
| 17 | 19.95040 | 26.99911 | 46.94951 |
| 18 | 11.97024 | 16.19947 | 28.16971 |
| 19 | 3.99008 | 5.39982 | 9.38990 |

*Y JPH ; Set number 4

| _AGE | YHARD | ysoft | Tot |
|---|---|---|---|
| 1 | 1.49405 | 2.22247 | 3.71652 |
| 2 | 6.69252 | 9.80962 | 16.50214 |
| 3 | 14.46825 | 22.23492 | 36.70317 |
| 4 | 23.14594 | 38.36357 | 61.50951 |
| 5 | 31.27294 | 56.99951 | 88.27245 |
| 6 | 37.79642 | 77.04647 | 114.84289 |
| 7 | 42.06663 | 97.56455 | 139.63118 |
| 8 | 43.77871 | 117.78408 | 161.56278 |
| 9 | 42.89683 | 137.09982 | 179.99665 |
| 10 | 39.57962 | 155.05617 | 194.63579 |
| 11 | 35.23946 | 155.24763 | 190.48709 |
| 12 | 31.53005 | 138.90577 | 170.43582 |
| 13 | 27.82063 | 122.56391 | 150.38455 |
| 14 | 24.11121 | 106.22206 | 130.33327 |
| 15 | 20.40180 | 89.88020 | 110.28200 |
| 16 | 16.69238 | 73.53835 | 90.23073 |
| 17 | 12.98296 | 57.19649 | 70.17945 |
| 18 | 9.27354 | 40.85464 | 50.12818 |
| 19 | 5.56413 | 24.51278 | 30.07691 |
| 20 | 1.85471 | 8.17093 | 10.02564 |

*Y HXS1 ; Set number 5

| _AGE | YHARD | ysoft | Tot |
|---|---|---|---|
| 1 | 2.56555 | 0.52150 | 3.08705 |
| 2 | 9.85870 | 3.63339 | 13.49209 |

```
   3    20.03295     9.89720    29.93015
   4    31.36011    18.93915    50.29927
  5    42.55321    30.04821    72.60142
  6    52.74985    42.42853    95.17838
  7    61.43464    55.32492   116.75956
  8    68.35531    68.08361   136.43892
  9    73.44855    80.17655   153.62509
  10    71.56026    81.55166   153.11192
  11    64.02760    72.96727   136.99487
  12    56.49494    64.38289   120.87783
  13    48.96228    55.79850   104.76079
  14    41.42962    47.21412    88.64374
  15    33.89697    38.62973    72.52670
  16    26.36431    30.04535    56.40965
  17    18.83165    21.46096    40.29261
  18    11.29899    12.87658    24.17557
  19     3.76633     4.29219     8.05852

*Y HXS2  ; Set number 6
_AGE YHARD  ysoft  Tot
   1     2.60785     1.57695     4.18480
   2    10.02484     8.23377    18.25862
   3    20.35790    19.45590    39.81380
   4    31.83625    33.51752    65.35376
  5    43.14569    48.67284    91.81853
  6    53.41054    63.49794   116.90848
  7    62.11237    76.96754   139.07991
  8    69.00262    88.42926   157.43188
  9    74.02559    97.54036   171.56596
  10    72.06363    96.11565   168.17928
  11    64.47798    85.99821   150.47620
  12    56.89234    75.88078   132.77312
  13    49.30669    65.76334   115.07003
  14    41.72105    55.64590    97.36695
  15    34.13540    45.52847    79.66387
  16    26.54976    35.41103    61.96079
  17    18.96411    25.29359    44.25771
  18    11.37847    15.17616    26.55462
  19     3.79282     5.05872     8.85154

*Y XSH1  ; Set number 7
_AGE YHARD  ysoft  Tot
   1     1.97098     1.11608     3.08705
   2     8.69462     4.79748    13.49209
   3    19.01216    10.91799    29.93015
   4    31.14814    19.15113    50.29927
  5    43.47508    29.12634    72.60142
  6    54.70345    40.47493    95.17838
  7    63.91006    52.84950   116.75956
  8    70.50633    65.93259   136.43892
  9    74.18494    79.44016   153.62509
  10    74.86206    93.12232   167.98438
  11    72.62284   106.76249   179.38533
  12    66.94697   107.83242   174.77939
  13    59.89992    96.48164   156.38156
  14    52.85287    85.13086   137.98373
  15    45.80582    73.78008   119.58590
  16    38.75877    62.42930   101.18807
  17    31.71172    51.07852    82.79024
  18    24.66467    39.72774    64.39241
  19    17.61762    28.37695    45.99458
  20    10.57057    17.02617    27.59675

*Y XSH2  ; Set number 8
_AGE YHARD  ysoft  Tot
   1     0.95373     2.92361     3.87734
   2     4.57882    12.41016    16.98898
   3    10.31480    27.00783    37.32263
```

| | | | |
|---|---|---|---|
| 4 | 17.09408 | 44.71061 | 61.80469 |
| 5 | 23.94636 | 63.71481 | 87.66118 |
| 6 | 30.14758 | 82.58453 | 112.73212 |
| 7 | 35.23243 | 100.26398 | 135.49642 |
| 8 | 38.95845 | 116.03670 | 154.99515 |
| 9 | 41.25619 | 129.46804 | 170.72423 |
| 10 | 42.17992 | 140.34628 | 182.52620 |
| 11 | 41.86494 | 148.62875 | 190.49369 |
| 12 | 39.23379 | 144.22703 | 183.46082 |
| 13 | 35.10392 | 129.04524 | 164.14915 |
| 14 | 30.97405 | 113.86344 | 144.83749 |
| 15 | 26.84417 | 98.68165 | 125.52582 |
| 16 | 22.71430 | 83.49986 | 106.21416 |
| 17 | 18.58443 | 68.31807 | 86.90249 |
| 18 | 14.45455 | 53.13627 | 67.59083 |
| 19 | 10.32468 | 37.95448 | 48.27916 |
| 20 | 6.19481 | 22.77269 | 28.96750 |

*Y BS1 ; Set number 9

| _AGE | YHARD | ysoft | Tot |
|---|---|---|---|
| 1 | 0.14943 | 2.00506 | 2.15450 |
| 2 | 0.64593 | 8.10306 | 8.74900 |
| 3 | 1.38567 | 17.30640 | 18.69207 |
| 4 | 2.23295 | 28.46884 | 30.70178 |
| 5 | 3.07340 | 40.58809 | 43.66149 |
| 6 | 3.82348 | 52.85990 | 56.68338 |
| 7 | 4.42916 | 64.67446 | 69.10361 |
| 8 | 4.86123 | 75.59413 | 80.45536 |
| 9 | 5.10983 | 85.32608 | 90.43590 |
| 10 | 5.17933 | 93.69486 | 98.87418 |
| 11 | 5.08388 | 100.61757 | 105.70145 |
| 12 | 4.84380 | 106.08214 | 110.92594 |
| 13 | 4.48273 | 110.12900 | 114.61173 |
| 14 | 4.02547 | 112.83595 | 116.86142 |
| 15 | 3.49646 | 114.30592 | 117.80238 |
| 16 | 2.91866 | 114.65726 | 117.57592 |
| 17 | 2.31289 | 114.01614 | 116.32903 |
| 18 | 1.69745 | 112.51075 | 114.20820 |
| 19 | 1.08791 | 110.26703 | 111.35494 |
| 20 | 0.49714 | 107.40556 | 107.90270 |

*Y BS2 ; Set number 10

| _AGE | YHARD | ysoft | Tot |
|---|---|---|---|
| 1 | 0.08762 | 2.71414 | 2.80177 |
| 2 | 0.48040 | 11.38187 | 11.86227 |
| 3 | 1.18387 | 24.63544 | 25.81931 |
| 4 | 2.12413 | 40.67311 | 42.79725 |
| 5 | 3.21124 | 57.89172 | 61.10296 |
| 6 | 4.36071 | 75.01826 | 79.37898 |
| 7 | 5.50172 | 91.11514 | 96.61686 |
| 8 | 6.57928 | 105.54239 | 112.12167 |
| 9 | 7.55382 | 117.90696 | 125.46079 |
| 10 | 8.39937 | 128.01145 | 136.41082 |
| 11 | 9.10137 | 135.80741 | 144.90878 |
| 12 | 9.65439 | 141.35535 | 151.00974 |
| 13 | 10.06003 | 144.79155 | 154.85158 |
| 14 | 10.32504 | 146.30147 | 156.62651 |
| 15 | 10.45973 | 146.09893 | 156.55866 |
| 16 | 10.47672 | 144.41023 | 154.88694 |
| 17 | 10.38985 | 141.46241 | 151.85226 |
| 18 | 10.21343 | 137.47488 | 147.68831 |
| 19 | 9.96161 | 132.65368 | 142.61529 |
| 20 | 9.64798 | 127.18784 | 136.83582 |

*Y JP1 ; Set number 11

| _AGE | YHARD | ysoft | Tot |
|---|---|---|---|
| 1 | 0.11162 | 1.22082 | 1.33244 |
| 2 | 0.40984 | 5.31606 | 5.72590 |

```
   3     0.78903    12.17494    12.96397
   4     1.14492    21.43331    22.57822
  5     1.39028    32.66681    34.05709
  6     1.45755    45.44910    46.90665
  7     1.29807    59.37683    60.67490
  8     0.88019    74.08105    74.96125
  9     0.18696    89.23202    89.41898
  10     0.00000   103.75408   103.75408
  11     0.00000   105.25756   105.25756
  12     0.00000    94.17782    94.17782
  13     0.00000    83.09807    83.09807
  14     0.00000    72.01833    72.01833
  15     0.00000    60.93859    60.93859
  16     0.00000    49.85884    49.85884
  17     0.00000    38.77910    38.77910
  18     0.00000    27.69936    27.69936
  19     0.00000    16.61961    16.61961
  20     0.00000     5.53987     5.53987

*Y JP2  ; Set number 12
_AGE YHARD  ysoft  Tot
   1     0.06937     4.62395     4.69332
   2     0.51292    19.79262    20.30554
   3     1.42203    42.27331    43.69534
   4     2.72101    67.92257    70.64359
  5     4.27398    93.37439    97.64837
  6     5.93613   116.30492   122.24106
  7     7.57898   135.33266   142.91165
  8     9.10097   149.81900   158.91997
  9    10.43001   159.66253   170.09254
  10    11.52154   165.12068   176.64222
  11    11.37233   158.03075   169.40308
  12    10.17524   141.39593   151.57117
  13     8.97815   124.76112   133.73927
  14     7.78107   108.12630   115.90737
  15     6.58398    91.49149    98.07546
  16     5.38689    74.85667    80.24356
  17     4.18981    58.22185    62.41166
  18     2.99272    41.58704    44.57976
  19     1.79563    24.95222    26.74785
  20     0.59854     8.31741     8.91595

*Y JPBS1  ; Set number 13
_AGE YHARD  ysoft  Tot
   1     0.27057     2.95134     3.22192
   2     1.19676    12.37632    13.57309
   3     2.54879    26.67663    29.22542
   4     4.02249    43.78590    47.80839
  5     5.37469    61.90074    67.27543
  6     6.44523    79.62394    86.06917
  7     7.14924    95.95957   103.10881
  8     7.46085   110.25970   117.72055
  9     7.39617   122.15830   129.55447
  10     6.99832   131.50683   138.50515
  11     6.32547   138.31716   144.64264
  12     5.44185   142.71383   148.15569
  13     4.41139   144.89542   149.30681
  14     3.97025   130.40588   134.37613
  15     3.52911   115.91634   119.44545
  16     3.08797   101.42680   104.51477
  17     2.64683    86.93725    89.58409
  18     2.20569    72.44771    74.65340
  19     1.76455    57.95817    59.72272
  20     1.32342    43.46863    44.79204

*Y JPBS2  ; Set number 14
_AGE YHARD  ysoft  Tot
1     0.29077     2.51246     2.80323
```

| | | | |
|---|---|---|---|
| 2 | 1.26841 | 10.54108 | 11.80950 |
| 3 | 2.70105 | 22.90734 | 25.60839 |
| 4 | 4.28331 | 38.02714 | 42.31045 |
| 5 | 5.76420 | 54.46523 | 60.22943 |
| 6 | 6.97002 | 71.05663 | 78.02665 |
| 7 | 7.79957 | 86.91873 | 94.71830 |
| 8 | 8.21084 | 101.42502 | 109.63585 |
| 9 | 8.20604 | 114.16608 | 122.37212 |
| 10 | 7.81801 | 124.90858 | 132.72660 |
| 11 | 7.09873 | 133.55694 | 140.65568 |
| 12 | 6.11032 | 140.11967 | 146.22999 |
| 13 | 4.91825 | 144.68088 | 149.59913 |
| 14 | 4.42642 | 130.21279 | 134.63922 |
| 15 | 3.93460 | 115.74470 | 119.67930 |
| 16 | 3.44277 | 101.27662 | 104.71939 |
| 17 | 2.95095 | 86.80853 | 89.75948 |
| 18 | 2.45912 | 72.34044 | 74.79956 |
| 19 | 1.96730 | 57.87235 | 59.83965 |
| 20 | 1.47547 | 43.40426 | 44.87974 |

*Y WSBF1 ; Set number 15

| _AGE | YHARD | ysoft | Tot |
|---|---|---|---|
| 1 | 0.60137 | 2.25313 | 2.85451 |
| 2 | 2.79884 | 9.92303 | 12.72187 |
| 3 | 6.29174 | 22.44432 | 28.73606 |
| 4 | 10.49933 | 38.64415 | 49.14348 |
| 5 | 14.86147 | 57.29784 | 72.15930 |
| 6 | 18.92321 | 77.29075 | 96.21396 |
| 7 | 22.35413 | 97.67382 | 120.02794 |
| 8 | 24.94177 | 117.67567 | 142.61744 |
| 9 | 26.57477 | 136.69513 | 163.26990 |
| 10 | 27.22293 | 154.28471 | 181.50765 |
| 11 | 26.91768 | 170.13065 | 197.04833 |
| 12 | 25.10831 | 168.46651 | 193.57483 |
| 13 | 22.46533 | 150.73319 | 173.19853 |
| 14 | 19.82235 | 132.99988 | 152.82223 |
| 15 | 17.17937 | 115.26656 | 132.44593 |
| 16 | 14.53639 | 97.53324 | 112.06964 |
| 17 | 11.89341 | 79.79993 | 91.69334 |
| 18 | 9.25043 | 62.06661 | 71.31704 |
| 19 | 6.60745 | 44.33329 | 50.94074 |
| 20 | 3.96447 | 26.59998 | 30.56445 |

*Y WSBF2 ; Set number 16

| _AGE | YHARD | ysoft | Tot |
|---|---|---|---|
| 1 | 0.68476 | 3.11343 | 3.79819 |
| 2 | 3.25682 | 13.67773 | 16.93456 |
| 3 | 7.29406 | 30.44630 | 37.74035 |
| 4 | 12.01278 | 51.30635 | 63.31913 |
| 5 | 16.70020 | 74.22844 | 90.92864 |
| 6 | 20.82458 | 97.51534 | 118.33992 |
| 7 | 24.04637 | 119.85691 | 143.90328 |
| 8 | 26.19311 | 140.31081 | 166.50392 |
| 9 | 27.22317 | 158.25456 | 185.47773 |
| 10 | 27.18924 | 173.32923 | 200.51847 |
| 11 | 26.20614 | 185.38409 | 211.59023 |
| 12 | 24.13470 | 180.76267 | 204.89737 |
| 13 | 21.59420 | 161.73502 | 183.32923 |
| 14 | 19.05371 | 142.70737 | 161.76108 |
| 15 | 16.51321 | 123.67972 | 140.19294 |
| 16 | 13.97272 | 104.65207 | 118.62479 |
| 17 | 11.43223 | 85.62442 | 97.05665 |
| 18 | 8.89173 | 66.59677 | 75.48851 |
| 19 | 6.35124 | 47.56912 | 53.92036 |
| 20 | 3.81074 | 28.54147 | 32.35222 |

*Y TL1 ; Set number 17

_AGE YHARD ysoft Tot

```
   1     0.86322     2.70307     3.56630
   2     3.30275     9.80337    13.10612
   3     6.21460    18.98790    25.20251
   4     8.72718    28.44761    37.17479
  5    10.39320    37.02548    47.41868
  6    11.08826    44.08591    55.17418
  7    10.88742    49.36543    60.25285
  8     9.96721    52.84707    62.81428
  9     8.53886    54.66460    63.20347
  10     6.80745    55.03363    61.84108
  11     4.95007    54.20422    59.15429
  12     3.10687    52.42997    55.53684
  13     1.37989    49.94905    51.32894
  14     0.00000    46.81021    46.81021
  15     0.00000    42.20050    42.20050
  16     0.00000    37.66510    37.66510
  17     0.00000    33.32221    33.32221
  18     0.00000    29.25091    29.25091
  19     0.00000    25.49896    25.49896
  20     0.00000    22.08988    22.08988

*Y TL2   ; Set number 18
_AGE YHARD  ysoft  Tot
   1     0.97878     3.21090     4.18968
   2     3.77982    11.69508    15.47490
   3     7.06603    22.48363    29.54967
   4     9.79509    33.27436    43.06944
  5    11.47135    42.66733    54.13868
  6    12.00186    49.96916    61.97102
  7    11.52759    54.97099    66.49858
  8    10.29496    57.76630    68.06126
  9     8.57268    58.61703    67.18971
  10     6.60486    57.86149    64.46635
  11     4.58928    55.85477    60.44405
  12     2.67147    52.93288    55.60434
  13     0.94783    49.39333    50.34117
  14     0.00000    44.96008    44.96008
  15     0.00000    39.68592    39.68592
  16     0.00000    34.67443    34.67443
  17     0.00000    30.02508    30.02508
  18     0.00000    25.79329    25.79329
  19     0.00000    22.00142    22.00142
  20     0.00000    18.64798    18.64798

;****Management costs $/ha****
*YT ?

;****Discount factors****
yDisc5% _DISCOUNTFACTOR(5%,10,half)

;****Revenue $/m3****
ytot$ 1 50

;****Costs $/ha****
yharvcost$ 1 5000

*Y HW1
 _AGE ysilv$
  1 18
*Y HW2
 _AGE ysilv$
  1 18
*Y HJP
 _AGE ysilv$
  1 88
*Y JPH
 _AGE ysilv$
  1 158
```

```
*Y HXS1
 _AGE ysilv$
  1 397
*Y HXS2
 _AGE ysilv$
  1 397
*Y XSH1
 _AGE ysilv$
  1 775
*Y XSH2
 _AGE ysilv$
  1 775
*Y BS1
 _AGE ysilv$
  1 775
*Y BS2
 _AGE ysilv$
  1 775
*Y JP1
 _AGE ysilv$
  1 156
*Y JP2
 _AGE ysilv$
  1 156
*Y JPBS1
 _AGE ysilv$
  1 311
*Y JPBS2
 _AGE ysilv$
  1 311
*Y WSBF1
 _AGE ysilv$
  1 775
*Y WSBF2
 _AGE ysilv$
  1 775
 *Y TL1
 _AGE ysilv$
  1 775
*Y TL2
 _AGE ysilv$
  1 775
```