EVOLUTIONARY AUTOMATA FOR VISUAL RESOURCE MANAGEMENT PLANNING AND HARVEST DESIGN

by

BRENT CHARLES CHAMBERLAIN

B.A. (Cum Laude), Pacific Lutheran University, 2002
B.B.A. (Cum Laude), Pacific Lutheran University, 2002

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

in

THE FACULTY OF GRADUATE STUDIES
(Forestry)

THE UNIVERSITY OF BRITISH COLUMBIA

April 2007

© Brent Charles Chamberlain, 2007
ABSTRACT

Increasing pressure to harvest in scenic vistas as a result of increased timber demand has caused significant changes to many beautiful natural scenes. Visual Resource Management aims to reduce the impact of harvests and improve their scenic design while trying to minimize the effect on timber availability. This thesis focuses on two main outcomes. First, a program was created that is capable of automating aspects of the design process in Visual Resource Management. The program, or Model, uses a modified genetic algorithm in combination with a geographical information system to create a final harvest plan that minimizes negative visual impacts for any given timber extraction level. The Model was tested across an array of different landscape terrain, including mountains, hills and valleys, to show its ability to deal with complex situations. Second, the thesis was created to better understand the relationship between timber availability and visible alteration. Results suggest a capacity for increased levels of aesthetic design while also increasing timber availability when compared with previous studies. Thus, the end product is a program that is capable of being adapted to real world situations by aiding in the harvest design process and producing a plan that tries to maximize both timber availability and the aesthetic properties of a landscape. This decision support tools allows planners to manipulate “what if” scenarios to ascertain the effects of varying timber extraction levels and visible alteration percentages.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Introduction</td>
<td>1</td>
</tr>
<tr>
<td>1.1</td>
<td>Thesis theme and objectives</td>
<td>3</td>
</tr>
<tr>
<td>1.2</td>
<td>Literature Review</td>
<td>4</td>
</tr>
<tr>
<td>1.2.1</td>
<td>Why is Visual Stewardship important?</td>
<td>4</td>
</tr>
<tr>
<td>1.2.2</td>
<td>Visual Resource Management</td>
<td>6</td>
</tr>
<tr>
<td>1.2.3</td>
<td>Creating and measuring the harvest using VRM</td>
<td>12</td>
</tr>
<tr>
<td>1.2.4</td>
<td>Tools to help orchestrate Visual Resource Management</td>
<td>15</td>
</tr>
<tr>
<td>1.2.5</td>
<td>Summary</td>
<td>22</td>
</tr>
<tr>
<td>2</td>
<td>Methods: Designing and Creating a Model</td>
<td>24</td>
</tr>
<tr>
<td>2.1</td>
<td>A Genetic Algorithm Approach</td>
<td>24</td>
</tr>
<tr>
<td>2.2</td>
<td>The Model Design</td>
<td>24</td>
</tr>
<tr>
<td>2.2.1</td>
<td>Terms</td>
<td>24</td>
</tr>
<tr>
<td>2.2.2</td>
<td>Background Information</td>
<td>24</td>
</tr>
<tr>
<td>2.2.3</td>
<td>Requirements</td>
<td>25</td>
</tr>
<tr>
<td>2.2.4</td>
<td>How the GA process is designed</td>
<td>26</td>
</tr>
<tr>
<td>2.2.5</td>
<td>Input</td>
<td>33</td>
</tr>
<tr>
<td>2.2.6</td>
<td>Output</td>
<td>38</td>
</tr>
<tr>
<td>2.3</td>
<td>The Model Apparatus</td>
<td>39</td>
</tr>
<tr>
<td>2.3.1</td>
<td>Description of Components</td>
<td>39</td>
</tr>
<tr>
<td>2.3.2</td>
<td>Steps</td>
<td>39</td>
</tr>
<tr>
<td>2.4</td>
<td>Measuring Perspective View</td>
<td>41</td>
</tr>
<tr>
<td>2.5</td>
<td>Scenarios</td>
<td>42</td>
</tr>
<tr>
<td>2.5.1</td>
<td>Hypothetical Scenarios</td>
<td>43</td>
</tr>
<tr>
<td>2.5.2</td>
<td>Realistic Scenarios</td>
<td>48</td>
</tr>
<tr>
<td>3</td>
<td>Results</td>
<td>52</td>
</tr>
<tr>
<td>3.1</td>
<td>General Outcomes</td>
<td>52</td>
</tr>
<tr>
<td>3.2</td>
<td>Scenario Outcomes</td>
<td>53</td>
</tr>
<tr>
<td>3.2.1</td>
<td>Hypothetical Scenarios</td>
<td>56</td>
</tr>
<tr>
<td>3.2.2</td>
<td>Realistic Scenario</td>
<td>69</td>
</tr>
<tr>
<td>3.3</td>
<td>Effect of Percent Visible Alteration on Timber Availability</td>
<td>70</td>
</tr>
<tr>
<td>3.3.1</td>
<td>Changing timber extraction requirements</td>
<td>70</td>
</tr>
<tr>
<td>3.4</td>
<td>Using Percent Visible Alteration Calculation</td>
<td>72</td>
</tr>
<tr>
<td>3.4.1</td>
<td>Scenario #3</td>
<td>73</td>
</tr>
<tr>
<td>3.4.2</td>
<td>Scenario #5</td>
<td>73</td>
</tr>
<tr>
<td>3.4.3</td>
<td>Scenario #6</td>
<td>73</td>
</tr>
<tr>
<td>4</td>
<td>Discussion</td>
<td>75</td>
</tr>
<tr>
<td>4.1</td>
<td>Timber Availability, VQOs and Empirical, but Hypothetical Data</td>
<td>75</td>
</tr>
<tr>
<td>4.2</td>
<td>VRM and the GA approach</td>
<td>78</td>
</tr>
<tr>
<td>4.3</td>
<td>Technical and Practical Issues with the Model</td>
<td>81</td>
</tr>
<tr>
<td>4.4</td>
<td>Future Application and Development</td>
<td>82</td>
</tr>
<tr>
<td>5</td>
<td>Conclusion</td>
<td>85</td>
</tr>
<tr>
<td>References</td>
<td></td>
<td>87</td>
</tr>
<tr>
<td>Appendix 1</td>
<td></td>
<td>100</td>
</tr>
<tr>
<td>Appendix 2</td>
<td></td>
<td>102</td>
</tr>
<tr>
<td>Appendix 3</td>
<td></td>
<td>104</td>
</tr>
<tr>
<td>Appendix 4</td>
<td></td>
<td>112</td>
</tr>
<tr>
<td>Appendix 5</td>
<td></td>
<td>132</td>
</tr>
</tbody>
</table>
LIST OF TABLES

Table 1.1 Adapted table of VQO descriptions ................................. 8
Table 1.2 Types of visualization software used by environment and resource planners, adapted from Lewis et. al. (2005) ........................................ 16
Table 2.1 Summary Table for GA Parameters Used ............................ 27
Table 2.2 Model's Numerical Input .............................................. 37
Table 2.3 Model's Non-Numerical Input ........................................ 37
Table 2.4 Hypothetical Scenario Variable Input Values ....................... 43
Table 3.1 Table structure used to report scenario results ................... 54
Table 3.2 Scenario #3 comparison of cell size and timber extraction levels ........................................ 59
Table 3.3 Scenario #6 Comparison of timber extraction levels ............ 66
Table 3.4 Comparison between both PVA measurement results for Scenario #3 ........................................ 73
Table 3.5 Comparison between both PVA measurement results for Scenario #6 ........................................ 74
LIST OF FIGURES

Figure 1.1 Theoretical short-term VQO-timber availability relationships .................................................. 11
Figure 1.2 Flow chart of a typical Genetic Algorithm .................................................................................. 20
Figure 1.3 Comparison of Genetic Algorithm vs. Random Approach ...................................................... 21
Figure 2.1 Diagram of GA heuristic used for model ..................................................................................... 27
Figure 2.2 Example parents to be bred ........................................................................................................ 28
Figure 2.3 Example crossover mask ............................................................................................................ 29
Figure 2.4 Example crossover using mask and parents ................................................................................ 29
Figure 2.5 Cases where constraint is violated .............................................................................................. 31
Figure 2.6 Cases where constraint is not violated ....................................................................................... 31
Figure 2.7 Minimum arrangement to meet constraint requirement ............................................................. 32
Figure 2.8 A Scenario DEM raster .............................................................................................................. 34
Figure 2.9 A Scenario Target Cut raster ..................................................................................................... 34
Figure 2.10 A Scenario Parent A raster ....................................................................................................... 34
Figure 2.11 A Scenario Parent B raster ....................................................................................................... 34
Figure 2.12 A Scenario Tree Height Map raster .......................................................................................... 35
Figure 2.13 A Scenario Visual Sensitivity Unit raster .................................................................................. 35
Figure 2.14 A Scenario High Contrast Surface raster .................................................................................. 36
Figure 2.15 A Scenario Harvest Plan and Modification Harvest initial input raster files .......................... 36
Figure 2.16 Scenario #1 DEM .................................................................................................................. 45
Figure 2.17 Scenario #2 DEM .................................................................................................................. 45
Figure 2.18 Scenario #3 DEM .................................................................................................................. 45
Figure 2.19 Scenario #4 DEM .................................................................................................................. 46
Figure 2.20 Scenario #5 DEM .................................................................................................................. 46
Figure 2.21 Scenario #6 DEM .................................................................................................................. 47
Figure 2.22 Scenario #7 DEM .................................................................................................................. 48
Figure 2.23 Scenario #1 DEM in 3D ........................................................................................................... 48
Figure 2.24 DEM for TFL 49 ..................................................................................................................... 49
Figure 2.25 Tree Height Map for TFL 49 ................................................................................................... 49
Figure 2.26 VSU for TFL 49 ....................................................................................................................... 50
Figure 2.27 High Contrast Surface for TFL 49 ............................................................................................ 50
Figure 2.28 Target Cut for TFL 49 ............................................................................................................. 51
Figure 3.1 Scenario #1 1.5m Best Solution ................................................................................................. 56
Figure 3.2 Scenario #1 41.5m Best Solution ............................................................................................... 56
Figure 3.3 Scenario #2 1.5m Vis Invisible Cells .......................................................................................... 57
Figure 3.4 Scenario #2 1.5m Best Solution ................................................................................................. 57
Figure 3.5 Scenario #2 41.5m Vis Invisible Cells ....................................................................................... 58
Figure 3.6 Scenario #2 41.5m Best Solution ............................................................................................... 58
Figure 3.7 Scenario #3 41.5m Best Solution ............................................................................................... 59
Figure 3.8 Scenario #3 41.5m 60% Timber Best Solution ............................................................................ 60
Figure 3.9 Scenario #3 41.5m 20m Res Best Solution .................................................................................. 60
Figure 3.10 High Mountain Face 3D render at 41.5m view point, 80% harvest ......................................... 60
Figure 3.11 High Mountain Face 3D render at 41.5m view point, 60% harvest .......................................... 60
Figure 3.12 High Mountain Face 3D render at 41.5m view point, 80% harvest, 20m cell ......................... 60
Figure 3.13 Scenario #4 1.5m Best Solution ............................................................................................... 61
Figure 3.14 Scenario #4 41.5m Best Solution ............................................................................................. 62
Figure 3.15 Scenario #4 High, Low and Average Visible Alteration Results for Each Generation 63
Figure 3.16 Scenario #5 1.5m Vis Invisible Cells ....................................................................................... 63
Figure 3.17 Scenario #5 1.5m Best Solution ............................................................................................... 64
Figure 3.18 Scenario #5 41.5m Vis Invisible Cells ...................................................................................... 64
Figure 3.19 Scenario #5 41.5m Best Solution ............................................................................................. 65
Figure 3.20 One High Hill Three Small Hills render at 41.5m view point .................................................. 65
Figure 3.21 One High Hill Three Small Hills render at 1241.5m view point ............................................ 65
Figure 3.22 Scenario #7 41.5m Vis Invisible Cells ...................................................................................... 67
Figure 3.23 Scenario #7 41.5m Best Solution ............................................................................................. 67
Figure A6.102 Scenario #7 1.5m Visible Cells ................................................................. 196
Figure A6.103 Scenario #7 Ground Vis Solution Progression ....................................... 197
Figure A6.104 Scenario #7 41.5m Vis Invisible Cells ...................................................... 197
Figure A6.105 Scenario #7 41.5m Best Solution ............................................................. 198
Figure A6.106 Scenario #7 41.5m Visible Cells ............................................................. 198
Figure A6.107 Scenario #7 Med Vis Solution Progression ............................................. 199
Figure A6.108 High Mountain Valley 3D render at 41.5m view point ............................. 199
Figure A6.109 Two High Mountain Valley 3D render at 1241.5m view point ............... 199
Figure A6.110 Scenario #7 41.5m Vis Invisible Cells ...................................................... 200
Figure A6.111 Scenario #7 41.5m Best Solution ............................................................. 201
Figure A6.112 Scenario #7 41.5m Visible Cells ............................................................. 202
Figure A6.113 High Mountain Valley 3D render at 41.5m view point ......................... 202
Figure A6.114 Two High Mountain Valley 3D render at 1241.5m view point ............... 202
Figure A7.1 Potential cell arrangement possibilities within Model .................................. 203
Figure A9.1 Viewshed Illustration .................................................................................. 209
Figure A9.2 Using the azimuth within the hillshade tool ............................................... 210
Figure A9.3 Using Angle within the hillshade tool ......................................................... 211
Figure A9.4 Hillshade calculation for Scenario #3, high resolution ............................... 212
Figure A9.5 Hillshade calculation times the visUgly to derive a weighted PVA .............. 212
Figure A9.6 VNS Render for surface analysis ............................................................... 213
ACKNOWLEDGEMENTS

It is with sincere appreciation and humility that I express my thanks to those who have encouraged, guided, prayed for and supported me throughout my life. I cannot express enough gratitude nor understand why so many have been so giving. If there is any portion of this document that should get the most attention and praise, it is right here.

To my friend Dr. Mike Meitner, thank you for your continued effort to establish a social and working relationship that is energized by curiosity and all things abstract. Thank you for your desire to serve as my advisor and committee chair, I look forward to producing more work together. Thank you to Dr. Shephen Sheppard for always having a pen and piece of paper ready, and for your heart-warming “cheers”. Thank you to Dr. John Nelson for your care and effort through my writing process.

Thank you to Dr. Sam Chung of the University of Washington - Tacoma, Dr. Chung-Shing Lee and Dr. Eli Berniker from Pacific Lutheran University for mentoring me and for planning the seed of the inevitable pursuit toward knowledge, discipline and learning. I would not be where I am today without your encouragement. Specifically, thank you to Dr. Eli Berniker and wife Frances Buckmaster for your friendship, artistic inspiration and beautiful stories. Thank you also to Dr. Carlos Ribeiro for sharing all your modeling expertise and most importantly all those chocolate truffles.

Thank you to the Forest Resource Management staff, Adelle Airey, Lucinda Ridgway, Marissa Relova, Heather Akai, Harry Verwoerd and Jerry Maedel for making the details in my life a whole lot smoother. Thank you to Dr. Ryan Gandy for many cups of coffee, memories in the lab and on the tennis court. Thank you to Julian Gonzalez for your friendship, hugs and influential character. Thank you to Patricia Masupyi for your smile and peaceful spirit. Thank you to Daniel Berheide for your wonderful conversations and calm demeanor. Thank you to Lisa Cashel for your hours of reading and editing every version of this document. Thank you to Ding “Klaus” Chen for your
diligence; for running the Model into the wee hours of the morning and for producing needed graphs and images.

To my friends from long ago and today, and to my many roommates, your words and actions have left an indelible mark on my heart. From whirlwind adventures, street-side songs, hours of prayer and late night conversations I would surely not be the person I am today without your friendship. To my extended family for your selfless desire to grow a child and encourage his potential, thank you.

To my mom for your love, to my dad for your service, to my sister for always being there for me and to my brother for continuing to build our relationship no matter how far apart we are. For all the times you have expressed your love in word and action; a true mark of hope. I love you all.

Lastly, to my wife Andrea, for over three years of courtship and marriage, thank you. You have influenced me to a degree I never knew possible. I admire your spirit, your desire for Christ's heart and your compassion. Thank you for picking me up when I have fallen, for the many breakfasts and selfless acts. Your character is an inspiration and joy. I love you.
DEDICATION

This thesis is whole-heartedly dedicated to my Lord, Jesus Christ,
the Father of all Creation, Truth, Hope, Peace, Joy, Love.

I would also like to dedicate this thesis to my late professor Dr. Jim "Jimmy Dale" Holloway

whose inspiration continues to stir moments of gratitude in my heart.

This thesis is also dedicated to my grandparents, great-grandparents and beyond. Your struggles, work and dedication to raising a family has trickled down through the generations.

Thank you for doing all you could to make my life full of opportunity.
1 Introduction

Super, Natural British Columbia. To some it beckons, to others it intrigues and to many it is home. British Columbia, Canada, is nestled between the deep blue waters of the Pacific Ocean and the towering peaks of the Rocky Mountains. Consisting primarily of the temperate deciduous forest biome, it is home to wild salmon, grizzly bears, orca whales, bald eagles and innumerable other kinds of wildlife and biodiversity. British Columbia spans a geographical region larger than Germany, Italy, and the United Kingdom combined, but BC only contains roughly 2% of those European countries combines population, leaving the vast land base with many available natural resources. For the residents who live here and for those who visit, the majestic beauty of the landscape can be beyond captivating and is never forgotten.

But this beauty is not invincible to change. The impacts of logging on the region’s natural scenic beauty are often substantial. Only recently has modern society changed the visual landscape on such massive scales that entire areas, like the Greater Vancouver Regional District (GRVD) and vistas surrounding it, have been transformed into completely different scenes. Aside from its local demand, the forest sector, BC’s largest industry, exports timber to numerous locations including the United States (US), China and Europe. The demand placed upon the industry to harvest at globally competitive prices has caused companies to seek the most efficient methods possible. The result: clear-cutting, the removal of all trees within a stand. Since the 1900’s, a patchwork-like landscape has developed, making certain forested regions appear similar to that of an irregular checkerboard. By the 1970’s, groups of people became so upset by what they saw that a “clear-cut crisis” began, leading to the creation of a visual resource management program in the US followed shortly thereafter by the implementation of a similar program in BC.

The duality of the public’s desire for scenic beauty and the global demand on forest resources leads to several questions. Can the conflict between the public’s desire for aesthetic quality and the demand on forest resources be alleviated? Are there any ways of mitigating the effect of timber harvesting on the visual quality of a forest vista? More specifically, is there a way to design a harvest plan that yields the highest level of visual quality given a specific timber extraction...
requirement? These questions pose a unique blend of social, technical and, at times, philosophical issues.

Countless researchers, landscape architects, harvest planners, foresters and others have focused a great deal of energy on finding solutions to these complex problems. A large body of research regarding public preferences for natural beauty and the factors effecting an individual’s perception about scenic beauty has accumulated in the last few decades (e.g. Daniel and Boster 1976, Daniel and Meitner 1997, Gobster 1995, Meitner et al. 2001, Miller 1984, Ribe and Matteson 2002). Technological advancements have made it possible to simulate landscapes and model potential solutions without having to alter any physical landscapes (e.g. (Bishop and Karadaglis 1997, Ervin 1992, Lewis et al. 2005, Meitner et al. 2003, Ribe 2005, Seely et al. 2004, Sheppard and Salter 2004). Combining these social, physiological and technical aspects of research, we are now well aware of those characteristics that cause the greatest negative reaction to harvesting.

Several harvesting techniques that reduce the visual impact have been proposed and some have been implemented on landscapes throughout BC. However, as of 2001, most logging in BC is conducted as clear-cut (Picard and Sheppard 2002). This may be due to several reasons, including: efficiency, opportunity cost in terms of safety and the "out-of-sight, out-of-mind" philosophy that hides the obvious like sweeping dust under a rug. Nevertheless, as demand for timber increases and the necessity for harvesting front-country in sensitive visual areas increases, companies will have to use alternative harvest solutions that meet the public’s preferences for scenic beauty.

Creating a harvest plan that minimizes the overall visual impact can be a difficult and arduous task even for the most experienced experts. Some recent research suggests that partial cutting in tandem with innovative design, can allow for timber extraction without causing major visual impacts (Picard 2002, Picard and Sheppard 2001a, Picard and Sheppard 2001b). A variety of harvesting techniques such as feathering, provide ways to mitigate the effects of visual degradation. However, these advanced techniques supplement proper harvest design
techniques, which together create a visually sensitive harvest design. Mountainous regions
blanketed with a lush dense forest cover that are easily visible from public areas such as parks,
roads and residences become complex puzzles for harvest plan designers to solve. Modeling,
research and trial-and-error have provided insight and knowledge of the effects of alternative
harvest solutions, but there is still much more to learn.

1.1 Thesis theme and objectives
The body of work presented in this document focuses on finding ways to alleviate the conflict
between aesthetic quality and the demand on forest resources. By incorporating technology,
experience and knowledge from a variety of disciplines, the end result of this research provides
forest designers with a tool that creates harvest plans that yield the highest level of visual quality,
based on perspective view visible alteration for a specific timber requirement. Emphasis is placed
on the creation of a model, titled the Visual Quality Harvest Designer (VQHD), which simulates
the design process and finds the “best-fit” solution for each scenario. A secondary emphasis, or
question, is whether or not the intensity of the established Visual Quality Objective infers a lower
amount of timber availability. The model’s measurement criteria will be guided by the British
Columbia Ministry of Forests’ (BCMoF) Visual Quality Objectives (VQO). The development of this
model, which automates aspects of the design process using core knowledge and modern
computer technology, will demonstrate innovative solutions to a variety of complex harvest design
problems.

The objectives of the research are six fold. First, to review if enough knowledge and information
exists to enable the development of an automated system capable of designing a harvest plan
that mitigates the visual impacts. Second, to test the feasibility of using a stochastic algorithm
(e.g. genetic algorithm) as an engine for the automation process. Third, to design a model that
automates aspects of the design process. Fourth, to test the model against a number of
hypothetical terrain examples which mimic potential harvest planning scenarios. Fifth, to discover
effect of the solutions derived from the model on timber availability. Sixth, to discuss the potential
for the model to be adapted as a decision support tool. The work in this thesis covers a range of
disciplines from forestry, to landscape architecture and computer science. The integration of these disciplines affords the unique opportunity to develop better tools and become better stewards of the visual landscape.

This document will address the first two objectives with a thorough literature review. The third objective will be dealt with in the Methods section. Several different landscape scenarios will be established and prepared for testing. The fourth objective will test the Model's ability to deal with the complexities in each scenario. The outcomes from these tests will be reported in the Results section. The fifth objective will also be wrestled with in the Results section and further addressed in the Discussion section. The sixth and final objective will be tackled in the Discussion. By the end of the five sections in this document, one should have a good understanding of the Model that was developed, its capabilities and potential for adoption into more mainstream realistic problems. One should also understand the implications in terms of the relationship between timber availability and visual quality.

1.2 Literature Review

1.2.1 Why is Visual Stewardship important?

In B.C., residents pride themselves on the natural beauty of their province. A cornerstone of B.C. tourism are the spectacular views (Sheppard et al. 2004a) and it has been demonstrated that places such as scenic byways generate large sums of tourism revenue (Sheppard et al. 2004b). Visual Stewardship is important because modifications of a local landscape can significantly and negatively effect property values (Iverson 1997). Not only are views important in an urban setting and for tourism purposes, but the proper care and management of scenic beauty is important to forest management as well. In 1974, the British Columbia Ministry of Forests (BCMoF) stated in the Visual Landscape Inventory procedures manual (B.C. Ministry of Forests 1974) that the province relies heavily on forest products and given the mountainous and rugged terrain, logging activity cannot simply be hidden from view. So, from an economic perspective, careful Visual Stewardship must be applied in order to limit the negative effects on property values and tourism.
Not only is the desire to manage forests with some visual care important in B.C., but it is also required by law (B.C. Ministry of Forests 2005). In 1960, 1969 and 1976 the United States issued three Acts (U.S. Congress 1960, U.S. Congress 1969, U.S. Congress 1976) that require scenic beauty and other intangibles to be considered in management decisions on National Forest land (Arthur 1977). Although this legislature was a first step toward environmental protection and management, implementing these goals in the context of modern society was left to others (Daniel 1990). One of the first major studies conducted within the framework of the 1976 legislature tried to find what about the forest caused such adverse reactions and what caused people to view a managed forest as acceptable (Daniel and Boster 1976).

Even today, the BCMoF’s Forest and Range Practices Act (FRPA) weighs the social aspects of forest operations. People seem particularly upset with certain kinds of forest management like clear-cutting (Dietrich 1992, Ribe and Matteson 2002), and this issue has become a political battle affecting many policy changes (Manning 1999, Steel et al. 1994, Yaffee 1994). Perhaps this is related to the public’s affective perceptions and emotional responses that influence attitudes and behaviors toward environmental policy and management (Grob 1995, Newhouse 1990). In cases where negative emotional responses are linked to forestry-related events such as old growth logging or clear-cuts, attitudes toward forestry can be affected negatively (Laird et al. 1982, Parsons et al. 1997, Tarrant 1999). The landscape, including the impacts of forest management, has been shown to impact one’s understanding of environmental health and ultimately one’s understanding of the environment in general (Greider and Garkovich 1994). Therefore, it makes sense that the BCMoF include aspects of management that care for the visual components of the landscape.

Forestry is a balancing act. Many ecological and economic factors must be integrated into plans for management of forest resources and have great influence on the forest industry in B.C. With all the complexities inherent in finding ways to mitigate the negative responses and visual impact of forest operations in visually sensitive areas, it becomes increasingly difficult to feed world-wide demand for forest products. Yet, thanks to years of experience, research and practical
knowledge, several methods have been developed to aid in the stewardship of visual resources while attempting to meet demand for timber.

1.2.2 Visual Resource Management

Visual Resource Management is one management approach under the umbrella of Visual Stewardship. It is a set of practices that typically aim to reduce the visibility of harvests and improve their scenic design, in order to increase the appearance of natural flow into the surrounding landscape (B.C. Ministry of Forests 2001, Bell 1993, Bell 2004, United States Forest Service 1977). Due in part to dwindling back-country timber availability (Picard and Sheppard 2002), companies are having to explore the possibility of increased forest operations in the front-country. The challenge facing these organizations is to find harvest systems that not only maximize visual quality but also, perhaps more importantly, minimize the "aesthetic dip" that tends to occur just after timber harvest (Ribe 1989, Ribe 2005, Sheppard et al. 2000). Finding harvest systems that minimize this immediate effect should help gain social acceptance for these new management practices and should mitigate negative public response that often is the true limit on these types of activities (Sheppard et al. 2000).

Although VRM has been practiced in the field for several decades, surprisingly little research has tested the effectiveness of different design techniques in reducing perceived visual impacts (Ribe 2005). A majority of this research is focused on the effect and impacts of clear cuts (Karjalainen and Komulainen 1999, Palmer et al. 1995, Pâquet and Belanger 1997, Ribe et al. 2002a). One study suggests that if a company wants to encourage public support of forestry, the organization would do well to avoid the use of clear-cut designs in visually sensitive areas and use alternative practices (Ribe 2005). As a management technique, VRM has successfully reduced conflict over landscape aesthetics in many different areas (Sheppard et al. 2000).

The BC Forest Practice Code Guidebook states that VRM was created to take visual values into account within resource management planning, forest operations and timber supply analyses. The guidebook provides a system for evaluating, assessing and monitoring changes within a
landscape. Three important steps within the system are the Visual Landscape Inventory (VLI), Analysis and Establishment of Visual Quality Objectives, and Planning and Visual Design (B.C. Ministry of Forests 2001). The purpose of VLI is to identify the characteristics, condition and sensitivity to alteration within the scenic area and is primarily concerned with areas that may be negatively impacted if managed without the application of the concepts, principles and practices presented in the Visual Landscape Management Process (B.C. Ministry of Forests 1997b). Furthermore the VLI process establishes a Visual Sensitivity Unit (VSU) which is a distinct topographical unit as viewed from one or more viewpoints and deemed visually sensitive to alteration. In addition its delineation is based on the homogeneity of the landform and of biophysical elements contained within it. Similar to "conducting an inventory" of company assets in a business, the VLI identifies visual resources (or assets). In B.C. the Ministry of Forests defines visual resource as the "quality of environment as perceived through the visual senses only" (B.C. Ministry of Forests 1997a).

After the Visual Landscape Inventory is completed and passed through the District Manager, final management recommendations, called a Visual Quality Objective (VQO), is established (B.C. Ministry of Forests 1997b, Picard 2007). A VQO is an objective that sets measurable levels for characteristics like visible scale, geometric shape of the area and allowable change in visible alteration (B.C. Ministry of Forests 2006). A particular VQO sets certain requirements which can be limited to guidelines such as allowable visible alteration. As a metric, the allowable visible alteration (the amount of allowable change in a landscape) is measured by the amount of contrast within the unit. The contrast is a measure of the relative dominance of natural versus human-modified areas within a landscape. In essence, a VQO sets visibility thresholds for human-caused visual change on the landscape (B.C. Ministry of Forests 1998b). The diagram below identifies the five VQO levels:
Table 1.1 Adapted table of VQO descriptions.
Adapted from Table 1 and Table 2 from (B.C. Ministry of Forests 2006)

<table>
<thead>
<tr>
<th>Visual Quality Class</th>
<th>Symbol</th>
<th>Basic Definition</th>
<th>Alteration percent of landform in perspective view</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preservation</td>
<td>P</td>
<td>consisting of an altered forest landscape in which the alteration, when assessed from a significant public viewpoint, is: 1. very small in scale, and 2. not easily distinguishable from the pre-harvest landscape.</td>
<td>0</td>
</tr>
<tr>
<td>Retention</td>
<td>R</td>
<td>consisting of an altered forest landscape in which the alteration, when assessed from a significant public viewpoint, is 1. difficult to see, 2. small in scale, and 3. natural in appearance.</td>
<td>0 – 1.5</td>
</tr>
<tr>
<td>Partial Retention</td>
<td>PR</td>
<td>consisting of an altered forest landscape in which the alteration, when assessed from a significant viewpoint, is 1. easy to see, 2. small to medium in scale, and 3. natural and not rectilinear or geometric in shape;</td>
<td>1.6 – 7.0</td>
</tr>
<tr>
<td>Modification</td>
<td>M</td>
<td>consisting of an altered forest landscape in which the alteration, when assessed from a significant public viewpoint, is very easy to see, and is: 1. large in scale and natural in its appearance, or 2. small to medium in scale but with some angular characteristics;</td>
<td>7.1 – 18.0</td>
</tr>
<tr>
<td>Maximum Modification</td>
<td>MM</td>
<td>consisting of an altered forest landscape in which the alteration, when assessed from a significant public viewpoint, is very easy to see, and is: 1. very large in scale, 2. rectilinear and geometric in shape, or 3. both.</td>
<td>18.1 - 30</td>
</tr>
</tbody>
</table>

The purpose behind establishing a VQO is to mitigate any drastic negative visual impacts on a sensitive unit by encouraging thoughtful design that meets the stated objective. According to Picard and Sheppard (2001b), Visual Quality Objectives are valuable as an effective performance standard in the front-country and increasingly in the backcountry where recreation and ecotourism occur (Picard and Sheppard 2001b). Establishing a VQO provides protection for the
visual quality of a region by creating a measurable metric that can be used as a legally binding regulation.

After a VQO has been established for a given region, the Planning and Visual Design process begins. During this step harvest plan designers attempt to find a plan that balances both aesthetic (VQO) and economic factors. The B.C. Ministry of Forests states that all the proposed operations within the design phase should consider economic, biophysical, ecological, and social values, in addition to creating a solution that meets visual resource guidelines and reduces negative visual impacts. This stage in development is necessary at both the landscape and stand levels. (B.C. Ministry of Forests 2001). Managing the innumerable elements and stakeholder desires, not to mention the challenges of nature, is not a simple task. Tools and training enable planners to develop effective plans.

1.2.2.1 Visual Resource Management and Effects on Timber Supply

Since VRM considers a balance between the intangible value of a scenic vistas and the economic value of timber supply, it may provide a win-win situation for both (Picard and Sheppard 2001b). In the last decade, a variety of companies and researchers have attempted to find ways in which to maintain or extend timber supply while reducing visual impact. For instance, a study conducted on a road corridor found that VRM increased timber availability and improved scenery by opening up views (McDonald et al. 1998). Using alternatives to clear-cutting may actually boost timber availability in visually constrained areas (Picard and Sheppard 2001b).

One alternative to the clear-cutting approach, called partial cutting, has been shown to increase timber availability and aesthetic quality when used instead of clear cutting (B.C. Ministry of Forests 1997a, Clay 1998, Fight and Randall 1980, McDonald et al. 1998). Partial cutting is not a silviculture system per se, but is often used in the same context. Partial cutting has been defined in several different ways (B.C. Ministry of Forests 1999, Dykstra and Heinrich 1996, Faculty of Forestry and the Forest Environment Lakehead University 2005, North Carolina Department of Environment and Natural Resources 2006, Wang and Pollack 1998), but the common denominator for all definitions is that partial cutting is not the same as clear-cutting, that not all
timber is initially harvested, and that it is not determined by scale as long as the cut is large enough to incorporate both cut and uncut trees.

Partial cutting affords a wide range of VRM opportunities because it is typically not tied with a set of silviculture rules in B.C. and it escapes the block adjacency requirements applying to clear-cutting. Picard and Sheppard (2001a) refer to partial cutting as an alternative to clear-cutting (Picard and Sheppard 2001a, Picard and Sheppard 2001b), but they do not limit which type of silviculture system to use. Partial cutting is referred to often in the remainder of this paper because the harvest method, and the flexibility it affords, has been found to be the most effective at mitigating the visual impact of the cut while increasing the total timber availability in visually constrained areas.

Several cases have been documented that use alternative approaches to either increase timber availability or reduce the overall effect on timber supply (Picard and Sheppard, 2001a; 2001b). In one example, the Robson Timber Supply Area (TSA) in BC used a combination of clear-cutting and partial cutting (Industrial Forestry Services Ltd. and B.C. Ministry of Forests 1998). The results of this TSA show that, over the long-term, timber supply will increase when partial cutting, rather than clear-cutting, is used in the more visually constrained areas (Picard and Sheppard 2001a). Another study, conducted on the Strathcona TSA, shows that in the areas subject to the partial retention VQO that partial cutting increased timber availability by 36%-46% (Timberline Forest Inventory Consultants Ltd. and Rowe 1999). This was a significant gain, considering that the area suitable for partial cutting was about 14%-25% of the TSA (Picard and Sheppard 2001a).

The diagram below, created by Picard and Sheppard (2001a), depicts a theoretical assumption of the correlation between VQO and timber availability. In this diagram, the authors have created a system that shows how partial cutting actually increases the timber availability.
Figure 1.1 Theoretical short-term VQO-timber availability relationships.

The concept presented above suggests that for any VQO, the highest amount of timber supply occurs when using partial cutting. In addition to increased timber supply, partial cutting also offers the highest visual quality when compared to clear-cutting (McDonald and Litton 1998). Picard and Sheppard (2001a) state that it is important to note the theoretical numbers presented in the figure above are preliminary hypothetical relationships based on two studies (B.C. Ministry of Forests 1996c, B.C. Ministry of Forests 1997a), but that the numbers are supported by several other related studies (e.g., Berris et al. 1989, Bradley 1996, Clay 1998, Pâquet and Belanger 1997) and much of their own practical experience. The authors also suggest that further studies should be conducted to test their theory.
1.2.3 Creating and measuring the harvest using VRM

For many years a harvest’s visual impact was measured by landscape architects, forestry professionals and other experts (Leopold 1969, Taylor et al. 1987). However, due to public outcry related to clear-cutting several different groups, such as psychologists, public land managers, lawyers and ecologists began researching what landscape and harvest characteristics affected the public’s reaction (Taylor et al. 1987). The knowledge gained from this research has enabled planners to become proactive in developing harvest designs that will minimize the visual impacts as judged by the general public.

1.2.3.1 Understanding the link between social preferences and biophysical properties of a landscape

In order to proactively create a harvest plan, correlations between the social aspects in terms of public preference and the biophysical characteristics of the harvest need to be understood. An expert who is asked to assess or develop a harvest plan with certain visual requirements uses several different elements of the landscape, such as texture, scale, form, edges and lines, position and movement and color of the landscape (Daniel 1990, Sheppard 2004). These elements and many of the perceptual relationships between a person and the landscape elements were developed in the late 1960’s by Litton (1968). Several techniques, including the Scenic Beauty Estimation method, have been created to measure an individual’s perceptual relationship to the biophysical properties of a landscape (Anderson 1981, Carls 1974, Daniel and Boster 1976, Daniel and Meitner 2001, Fairweather and Swaffield 2000, Heft and Nasar 2000, Newton et al. 2002, Ribe 1999, Shafer and Richards 1974).

Most of the literature concerning preference judgments, deal with the effect of clear-cuts. For instance, most researchers would agree that evidence leans toward clear-cuts as not being preferred and that any type of disturbance is worse than no disturbance (Gobster 1995). When synthesizing results from the studies conducted on the public’s preferences, findings have overwhelmingly shown that people prefer uncut to clear-cut forests, that higher human-made contrasts in the landscape are generally less preferable and that disruption is generally found as
less beautiful but sometimes still acceptable (Miller 1984, Palmer 1998, Palmer et al. 1994, Paquet 1993, Paquet and Belanger 1997, Ribe 1999, Ribe 2005, Ribe et al. 2002a). Additionally, the USDA recognizes that practices should minimize clear-cut size and undulate the edges of openings to make them less noticeable or less regular and intrusive; and therefore more preferable (USDA Forest Service 1974). For the most part, the more disruption of a forest that is perceived by an individual, the lower the visual preference will be rated (Ruddell et al. 1989).

The relationship between clear-cuts and public preference is one of strong antipathy. In response, other researchers wanted to explore what methods would actually reduce the initial impact of seeing a clearcut. Strip clear-cutting was proposed and was found to provide potential reductions for negative public reactions toward logging (McDonald and Stokes 1997). Visual buffers may also reduce visual impact of logging activities (Karjalainen and Komulainen 1999). Concerning silviculture methods, uneven-aged management tends to have less visual impacts than even-aged management (Hull 1988). In the case of the shelterwood system, although it does have an initial loss of visual quality, the initial loss is still less pronounced than a clear-cut (Ribe 1991). Also, using smaller areas for clear-cuts and making clear-cuts more "organically" shaped reduces the overall visual impact (Hull 1988).

1.2.3.2 Measuring the harvest design for effectiveness

In areas specified for VRM, staying within the established VQO cannot guarantee public acceptability, but does provide some structure with which to mitigate the overall impact. The hope is that the VQO, as measured by the allowed percent of visible alteration, will meet with the desires of the public. One study found a high degree of correlation between VQOs and public preferences (Berris et al. 1989). In regards to specific findings related to VQOs, another key study conducted by the BCMoF suggested that people prefer more "natural" appearing conditions on a landscape, reflecting acceptability for Preservation and Retention level VQOs (B.C. Ministry of Forests 1996c). The same study showed high levels of unacceptability with Maximum Modification VQO (B.C. Ministry of Forests 1996c), so public acceptability can not be guaranteed when following VQO guidelines.
Assessment of the harvests could be done in a number of fashions. Scenic beauty assessment and public acceptability are valuable methods used to gauge how the public feels. Measuring the difference between what was planned and what actually happened (e.g. was the final shape consistent with the planned shape) could also be completed. Yet, these methods do not quite reach a level of scientific rigor that expresses exactly what about the cuts (in regards to color, shape and pattern) really affects the public's preference. If there is any correlation between scenic beauty and formal aesthetic landscape features, would that not be beneficial for future planning?

Research that attempts to find correlations of this type is limited, but initial studies have been carried out. The complexity in finding visual characteristics and their thresholds of influence offers a hint as to why few studies have been conducted (Shang and Bishop 2000), let alone if it is even possible (Carlson 1977). Still, the potential exists that these characteristics can be measured through the psychophysical approach. One important body of work found that measuring in square minutes of visual angle as the proportion of the field of view occupied by an object, multiplied by the percentage contrast created a metric called contrast weighted visual size (Shang and Bishop 2000). By developing this metric and testing it, the authors discovered that it is not only the contrast difference or the size of the contrast alone, but it is the combination of both over a landscape that has the largest effect to viewers. Other authors have contributed knowledge and research that provides ways to measure the impact on visual changes in the landscape (Hopkins 1971; Iverson 1985). A more methodological approach, called GEOptics, uses 3D imaging technology to determine the illumination intensity from a 3D surface by measuring the amount of light reflectance of a surface (Fairhurst 2006). All these approaches provide metrics and methods to measure the amount of change in the landscape as a result of modifications.

In British Columbia, the BCMoF uses a set of procedures that attempt to measure the effectiveness of a particular cut to meet a VQO (B.C. Ministry of Forests 2006). This set of procedures placed emphasis on measuring the amount of "percent alteration." Percent alteration is defined as the scale of disturbance to a landscape as a result of human activity (e.g. cutblocks)
and is expressed as a percent of the total scene visible, or Visual Sensitivity Unit (VSU) in perspective view (B.C. Ministry of Forests 1997b). The "percent alteration" is the metric that will be used throughout this thesis to derive an effectiveness measurement. This metric has similarities to the Shang and Bishop (2000) study in that it focuses on the amount of contrast within a VSU.

1.2.4 Tools to help orchestrate Visual Resource Management

Several tools will be assessed to determine the feasibility of creating a model that can automate the harvest design process by using the measurement techniques and planning methods described in the previous sections of this chapter. As a point of clarification within this thesis, terms like automation, simulation, and depictions all fit under the umbrella of modeling. Technically, modeling can include one or more tools such as custom code, algorithms, frameworks and software programs. Conceptually, modeling uses a set of rules and parameters which help produce some sort of image, plan or set of information as outcomes. Both the conceptual and technical perspectives will be elaborated upon in order to form an understanding of how they play an integral role in VRM through decision-support means.

1.2.4.1 Visualization Tools

A wide variety of software tools are available to aid the VRM process (Lewis et al. 2005, McGaughey 1998). These tools include software programs that create photo-realistic images, produce plan view maps and edit photographs. Most of the software is focused on providing visualizations, which can be defined broadly as "...pictures of objects, conditions, processes, or places that help the viewer understand and interpret the subject matter by revealing its appearance or visually displaying certain significant characteristics" (Lewis et al. 2005). Lewis et al. (2005) provides a table that describes the range of visualization types used by environmental planners and resource managers. McGaughery (1998) provides a review on several different types of visualization techniques, including geometric modeling, video-imaging, geometric video-imaging and image draping, which were used around 1998 to depict the effect of forest harvesting operations.
Table 1.2 Types of visualization software used by environment and resource planners, adapted from Lewis et. al. (2005).

<table>
<thead>
<tr>
<th>Technique Description</th>
<th>Software Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Geographic Information System (GIS) or Computer Aided Design (CAD) software</td>
<td>AutoCAD©, MicroStation© or any GIS software (e.g. ArcView, ArcInfo or ArcGIS).</td>
</tr>
<tr>
<td>mathematically interpolates a frame to depict an elevation model.</td>
<td></td>
</tr>
<tr>
<td>Simulated changes to a landscape are manually composed using photo-editing software.</td>
<td>AutoCAD© or MicroStation©, GIMP and Adobe PhotoShop.</td>
</tr>
<tr>
<td>Sketches and drawings could also fit into this category.</td>
<td></td>
</tr>
<tr>
<td>Photo-realistic simulations build images based on underlying data. There is no manual</td>
<td>World Construction Set, Visual Nature Studio or GIS Software.</td>
</tr>
<tr>
<td>editing, except to ensure that the data is configured properly.</td>
<td></td>
</tr>
</tbody>
</table>

Over the last 30 years, the progression of manually produced graphics toward computer-generated visualizations has realized a significant advancement and increased uses, in regards to environmental visualization (Sheppard and Salter 2004). A few research articles highlight the use of visualization in the context of environmental management and forestry related resource management (Bishop and Karadaglis 1997, Orland 1992, Sheppard 1989). Two dimensional visualizations, such as maps and GIS inventory layers, are fairly common in forestry (Meitner et al. 2005a). Several two dimensional visualization techniques have been used to understand a variety of problems and make recommendations for solving them (Ribe et al. 2002b, Wing and Johnson 2001). Another commonly used visualization technique deals with simulations in the third dimension. Software that develops three dimensional forestry visualizations have been around since the late 1970's. Programs such as Perspective Plot (Twito 1978) and PREVIEW (Myklestad and Wagar 1976) were some of the first software programs developed that used digital terrain models and simple visuals to depict clearcuts on a landscape. GIS applications that drape orthophotos on top of digital elevation models can also be viewed in 3D (Bergen et al. 1995), using a GIS program such as ESRI’s ArcScene. In recent years, high detailed photo-realistic visualizations have become an integral part of community planning and resource management applications (Chamberlain et al. 2005, Lewis and Sheppard 2005, Lewis et al. 2005, Lewis et al.
Meitner et al. 2003, Meitner et al. 2005a, Meitner et al. 2005b, Seely et al. 2004). Several other programs and implementations of three dimensional software solutions that help planners manage visual resources have been used in the past (Lim and Honjo 2003, Tyrvainen 1999).

Both two dimensional and three dimensional visualization tools provide a simple way of showing forest planners and stakeholders the past, present and potential future across a range of realism within depictions. The tools provide a method that allows planners to play “what-if” scenarios to see what visual effect may occur as a result of certain plans (Ervin 1992). Furthermore, the visualization software has become more and more complex, allowing for automation of visualizations. By integrating this software with management plans, planners can have quick feedback about the potential result of implementing their plan.

1.2.4.2 Algorithms

As complex as the visualization software is today, it is not tailored to design harvest plans that minimize the visual impact. This process has been produced by human interaction with the software. The desire then, is to create a program using code and an artificial intelligence engine that can produce a harvest plan that can then be easily visualized in two or three dimensions. A variety of programming languages exist that provide a platform for development. The type of problems that use an artificial intelligence engine are usually stochastic loops, sometimes called meta-heuristics, whereby through each iteration of a loop, the program incorporates new information and progresses toward a better understanding of what the desired outcome should be.

Several recent implementations of stochastic loops have been used within forestry, employing a few of the well known search techniques such as simulated annealing (SA), tabu search and genetic algorithms (GA) (e.g. Boston and Bettinger 1999, Boston and Bettinger 2002, Brumelle et al. 1998, Crowe et al. 2003, Falcao and Borges 2001, Lu and Eriksson 2000, Weintraub et al. 1994, Weintraub et al. 2000). The purpose of using these techniques is to find a good or near optimal solution for a particular problem. The three techniques are different from one another in the way they search the solution space. Several modifications or adaptations of the three different
techniques can be implemented in order to address a variety of forestry related problems. Simulated Annealing is an approach that takes a potential solution and modifies it at random to find a better solution (Boston and Bettinger 1999, Laarhoven and Aarts 1987, Murray and Church 1995, Nelson 2003). Tabu search is another search technique but differs from SA by the way it explores the solution space. The tabu search uses a local neighborhood approach where solutions are derived by fixating on a problem locally and continually modifying the structure of the neighborhood to ensure that unexplored search spaces are assessed (Glover 1989, Glover 1990). The genetic algorithm is based on evolution theory and has been developed over the last five to six decades from an area of research called “evolution strategies” discovered in the 1960’s (Rechenberg 1965). This algorithm uses an approach that simulates the breeding process in nature. In this approach, a solution is designed in terms of a chromosome, which is generally a string containing values. Several chromosomes are initially generated randomly and are called the population. Through evolutionary processes, these chromosomes are bred (crossover), mutated and assessed for their fitness value. Those chromosomes that have the highest fitness value are then selected for breeding in the next generation. A continuum of breeding and subsequent assessment for fitness values is used to eventually converge upon a “best fit” or “best” solution (Falcao and Borges 2001). The genetic algorithm and simulated annealing are quite similar in concept. Both include assessment criteria and some sort of random mutation. However, the fundamental difference is that the genetic algorithm has a population of potential solutions that allow for a crossover function to exist, whereas simulated annealing uses one potential solution and therefore no crossover can take place.

The model presented in this thesis was based upon a genetic algorithm (GA) approach. The typical GA approach uses three primary variables as input: (1) population size, (2) crossover rate, and (3) mutation rate (Boston and Bettinger 2002, Falcao and Borges 2001). Population size is the number of offspring that will be produced in each generation. Crossover rate determines the percent of a solution that will be carried from one generation to the next. So, a crossover rate of 80% would mean that 80% of one “parent” solution and 20% of another “parent” solution would
be copied to form a new "child" solution in the next generation. Mutation rate is the percent of the child solution that will be altered.

The GA approach also uses two key components that must be established before executing the algorithm: (1) an initial population and (2) the fitness parameter (Goldberg 1989). The initial population is defined by at least two parent solutions. The fitness parameter is a metric that assesses each child solution for its viability as parents for the next generation. The fitness parameter can be made from a complex set of variables and computations or a simple measure using one variable. The idea behind the fitness parameter is that it attempts to find a "best fit" within a given problem. In the physical world, this can be represented by the natural selection process where one male (alpha male) becomes the only male breeder within a group. However, unlike the natural process where one male and one female reproduce, the GA does not consider the gender of a child, but rather just the fitness value.

The typical GA operations can be described in this step-by-step process (shown in Figure 1.2) as presented below:

1. Select initial population of solutions
2. Determine the fitness for each solution
3. Choose the best solutions based on fitness
4. Breed the new generation through crossover of parents and mutation
5. Determine the fitness for each new solution
6. Loop to #3 or terminate if solution is found
Figure 1.2 Flow chart of a typical Genetic Algorithm

For more detailed information about how the GA generally functions, please refer to Appendix 1.

In terms of results, the GA attempts to increase the likelihood of reaching certain fitness levels after each generation. By breeding the best solutions and mutating them slightly over time, the genetic pool becomes more fit. However, much like natural processes, the mutations and potential offspring could result in solutions that are less fit than their parents. This progression toward fitness can be illustrated in the diagram below, which compares the progression of the GA versus a random search method to illustrate how the GA is better at finding a better solution than the random approach (better is determined by a lower value on the Y axis). Figure 1.3 uses the average of ten children within each generation's population. Figure 1.3 uses a similar approach as Venema et al. (2005) to depict the difference between random search algorithms and GA search algorithm, but this particular figure is specific to the current problem. Figure 1.3 uses
population data generated by the GA heuristic explained in Chapter 2, the data come from Appendix 6.1.6 (80% harvest rate). The Y axis shows the percent visible alteration in perspective view from the average of ten solutions. The X axis shows a hypothetical progression of the average values as the algorithm iterates.

Figure 1.3 Comparison of Genetic Algorithm vs. Random Approach. A lower visible percent alteration is considered a better or more fit solution.

As is depicted by the figure above, a trend toward a more fit solution occurs over time. Fluctuations in fitness levels for each generation in the GA occur by modification and crossover. It is these fluctuations that encourage the adaptation toward more fit offspring over time. In the random approach, more significant fluctuations occur, but none arrive remotely close to the levels attained by the GA approach.

The Genetic Algorithm has been used in several forestry applications. One unique approach that used the GA to solve land use management problems tried a goal-based approach, wherein planning constraints were considered “fuzzy” (Stewart et al. 2004) meaning that exact
requirements were not established. The authors used an algorithm to identify a two land use
designations, find two parents and select half of each designation from them. These allocations
were then used as crossover for the development of a new child. The heuristic they developed
mutated the solutions by creating a random selection of cells within a randomly-sized cluster and
changing the values of those cells. This process was continued until the weighted average of the
goodness of fit for each goal was at its highest level. Another use of the GA within forestry is the
integration of the algorithm with harvest scheduling problems (Boston and Bettinger 1999, Falcao
and Borges 2001). In the Boston and Bettinger (1999) method, some heuristics were used to
adapt the GA for their specific problem, including a technique that converted the two dimensional
problem into a single dimensional array that was then manipulated by a standard GA crossover
and mutation. Another harvest schedule algorithm used a modified GA to produce a harvest plan
for a 70 year cycle (Falcao and Borges 2001). This study used the GA by assigning values to a
grid of cells, each value representing a 70 year management plan. The fitness value was
determined by determining constraints, such as adjacency, and ensuring the maximum harvest
over the entire 70 year duration. Creating harvest units using raster based files is yet another
implementation of the GA that has been developed (Lu and Eriksson 2000). The premise of the
algorithm was to design harvest units on an area encompassing 100x100 cells at a resolution that
simulated 400ha. The purpose was to meet the demand of configuring smaller harvest units
within operational planning for the entire site. The use of the GA to produce harvest units can also
be applied to designing a patch configuration within a stand (Brookes 2001). Evaluations of the
final solutions from this study were completed using an integrated operation with FRAGSTATS
(McGarigal and Marks 1995) that measured size, shape and patterns of the patches.

1.2.5 Summary

A range of tools, measurement systems and knowledge are available to help mitigate the
negative impacts of visual resources through proper management. There are good reasons why
VRM should be applied to forestry practices, and applied research has shown that it is feasible to
incorporate VRM guidelines. Recent developments in technology can provide a way to model the
future effects and assess the outcomes of current harvest plans. Even with all of the tools and
information available, however, no implementation of an automated system that begins at design stage and ends with potential harvest solutions that meet visual requirements is available. Furthermore, the complexity of the planning process, wherein a design has to meet certain visual requirements, can pose a difficult and arduous task. If any of the mundane or complex tasks could be automated using technology, this could afford planners more time to focus on key elements and creative implementation of harvest plans that meet visually sensitivity requirements.
2 Methods: Designing and Creating a Model

2.1 A Genetic Algorithm Approach

The model presented in this thesis was based upon a genetic algorithm (GA) approach. The GA approach is rooted in evolutionary theory and can be considered a form of Artificial Intelligence. The algorithm uses computational methods to model the evolutionary process given a defined world. In the context of this thesis, the defined world is a specific scenario that requires a solution to a problem. Generally, the GA is used to help explain the evolutionary process such as species development and change through breeding, but it can also be used find answers to complex problems. For the purposes of this thesis, when referring to the word "solution", the word is used to represent a potential result to a given problem.

2.2 The Model Design

2.2.1 Terms

For this section, the term 'model' will refer to an ArcGIS model developed in ESRI's Model Builder platform using tools from the ArcInfo version. Using the term 'Model' with capital letters or 'program' or 'Program', refers to the Model in its entirety. The reason for the distinction is that the Model incorporates several processes and sub-processes, tools and scripts that ESRI refers to as models. Separating the two terms will help keep references to ESRI's Model Builder and related components, and references to the program or Model generated for this thesis as clear distinctions.

2.2.2 Background Information

The intent of the model was to automate the basic design of forest harvesting when constrained by visual quality objectives, a process that currently is manually conducted by an expert. Unlike the expert approach, an automation process has the ability to play "what-if" scenarios and discover the effect of minor modifications over an entire landscape in perspective view. It would be an arduous and lengthy task for an expert to understand how small changes impact the overall visual appeal. The reason for this is that human beings find it quite difficult to take a highly complex three dimension problem and translate it onto a two dimension map. If this translation
process can occur via an automated process, it would allow harvest planners to focus on fine
tuning the specifics of the design while allowing the computer to direct the designer towards a set
of alternatives that may not have been readily apparent. The Model that was created fulfills these
objectives by automating the harvest design process while requiring little user input or
supervision. This section details the inputs, outputs, requirements, constraints, limitations and
process of the Model.

The platform for development was a combination of ArcInfo's Model Builder and custom code
written in the Python programming language. ArcInfo's Model Builder allows for the development
of a program based on a modeling approach. ArcInfo also provides a number of tools such as
viewshed and map algebra. One of the most important reasons for choosing this software
package is that it has become the industry standard in terms of raw geoprocessing and has a
number of integrated tools built into the system making both development and knowledge sharing
easy. Many of the tools in the software have been developed by statisticians, mathematicians,
physicists, computer scientists and others who are experts in their field. Although access to the
internal logic and algorithms are limited, documentation about these details is provided by ESRI.

There are many advantages to this modeling approach. The major advantage realized for this
thesis was the speed at which model development could take place. Most of the structure for the
model took less than two weeks to create. This design phase included not only a flow diagram of
the process, but also a running model that generated scenarios as raster files. Subsequent
changes and additions were made to the order of processes and the type of output needed.
These changes required no coding for the flow and were easy to make because all of the
organization of code was managed by ArcInfo in the background.

2.2.3 Requirements

Automating the creation of a harvest plan that reduces overall visual impacts, requires defining
how to measure visual impact. As mentioned in the literature review, there are several ways that
visual impact can be defined. The literature generally agrees that a combination of contrast and
the amount of contrast on a landscape, as a result of human modifications, tend to have the most
significant impact on visual quality (Shang and Bishop 2000; BCMoF 1997; Ribe 2005; Ribe 2002). Therefore, the Model will use a weighted contrast measurement, which measures the percent of change in visible alteration from perspective view, to compare outcomes between each child within a generation. Change in visible alteration can be explained in the following example: if a clear-cut creates a hole in the green forest cover causing a break in the contrast (green to brown), that area would be considered a negative change in perspective alteration. The BCMoF provides a resource detailing how to measure the viewable alteration on a landscape (B.C. Ministry of Forests 2006). In terms of the visual impact measurement, the requirement is to determine the amount of alteration where any human-induced change as a result of the harvest, produces additional contrast to the forest cover that has a negative impact (e.g. pre-VEG state or surface with soil colors) within a Visual Sensitivity Unit (VSU). Therefore, the model is trying to find a harvest solution that reduces visual impact for a given timber harvest rate within a VSU. This does not mean that the GA will produce the best possible or optimal solution, but it attempts to quickly produce a solution that meets all user requirements.

2.2.4 How the GA process is designed

One of the primary challenges of automating the harvest planning process to meet the requirements specified in Section 2.2.3 is to create a method which can develop a plan in two dimensions and accurately measure the potential impact of that plan on a three dimensional surface. Integrating a method that can develop in plan view and measure the effects in perspective view is managed in the presented Model by using the standard, selection, crossover and mutation processes in the GA with some unique heuristics and measurement techniques. The Model uses a raster based format to efficiently modify solutions in 2D for each generation, then uses the same raster draped onto a 3D surface to measure the visible effects. If the perspective view or 3D visible effects have been measured and meet the user requirements, then a solution has been found. The GA heuristic for this Model is explained conceptually in the following sections. Table 2.1 provides the summary table of how the GA parameters are implemented in the Model presented. Figure 2.1 is a flow chart of the processes within the GA heuristic used for the presented Model. The design details are more thoroughly explain in
Appendix 4. The fitness measurement technique used to provide a percent alteration in perspective view based on a 2D plan is detailed in Appendix 9.

### Table 2.1 Summary Table for GA Parameters Used

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Summary Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Selection</td>
<td>Best two solutions</td>
</tr>
<tr>
<td>Crossover</td>
<td>Uses a randomly created 2 dimensional mask which is applied to the selection population</td>
</tr>
<tr>
<td>Mutation</td>
<td>Uses a randomly created 2 dimensional map as 1/3 spatial resolution (e.g. if the solution resolution is 20m, then the mutation map is produced at 60m).</td>
</tr>
<tr>
<td>Completion</td>
<td>If solution meets Allowable Visible Alteration or the maximum number of iterations has been run</td>
</tr>
</tbody>
</table>

![Diagram of GA heuristic used for model](image)

**Figure 2.1 Diagram of GA heuristic used for model**

2.2.4.1 Selection

One of the limiting factors of operation, the length of time it takes one generation to be computed, required that the selection process be streamlined in order to save time. One way the GA can be accelerated toward a solution is to draft certain rules for the selection process. Some
implementations of the GA allow for a random mutation between any child or a certain selection of children. In the case of the Model presented, only the two solutions with the lowest visible alteration for each generation are selected as parents. This selection process decreases the time it takes to diverge upon a particular solution, or genetic sequence, but it could have an effect of limiting the variety that a bred in each generation. Meaning that it could affect the potential to reach a global optimum and stop within a local optimum. However, the effect could be reduced by altering the rate of crossover and mutation. In the case of this model, upon evaluation of all children, the two solutions with the lowest visual impact are selected and bred in the next generation.

2.2.4.2 Crossover

The method used in this Model considers a crossover to be any grouping of any length of values within a genetic sequence. The grouping can be ordered in any manner or direction, so linearity is not a requirement for crossover. By using a different mask for each creation of a child from two parents, any order of genetic crossover can occur. For instance, take the following two parents in Figure 2.2 where black represents 1 and white represents 0:

![Parent A](image1)

![Parent B](image2)

**Figure 2.2 Example parents to be bred**

Then, a crossover mask is randomly generated by assigning a value of 0 or 1 to each cell. A mask might look like the grid below.
Figure 2.3 Example crossover mask

The mask simulates an 80% crossover rate, so it assigns 80% of the cells to a value of 1. Then, by extracting from Parent A, where all values in the mask equal 1, and extracting from Parent B, where all values in the mask equal 0, a child solution is derived that may look like this:

![Example crossover mask](image1)

Figure 2.4 Example crossover using mask and parents

In terms of crossover, the employment of a mask simulates an approach with no direction bias. This method seems to have a higher degree of fidelity to the situation than converting the two dimension problem into a single dimension scenario as is the standard GA approach depicted in Appendix 2.

2.2.4.3 Mutation

The mutation step is completed much in the same way as the crossover, by using a mask. The mask used for the crossover manages a smaller cell resolution (instead of a 9x9 grid it would works within a 3x3 grid) than the resolution for either of the parents. The reasoning behind this is to encourage the potential for a random cluster to be started. Since the constraint requirements require a specific arrangement of no less than three cells, using a mutation at the same spatial resolution as the parents, would likely never result in a mutation cluster developing. For instance, if one cell is mutated as a harvested cell, but exists away from any other cells, then that cell would be removed from the harvest plan as a result of the cell adjacency constraint performed at a later step. This is because a mutation would have to be randomly generated in such a format.
that three cells would be selected to meet the constraint requirements. By reducing the resolution by a factor of three, the likelihood of a new cluster existing after a mutation increases.

The mutation mask is generated by reducing the parent resolution by a factor of three. Then a random value between 0 and 1 is assigned to each cell. If a cell value falls below the mutation rate, it is assigned a value of 1. If a cell value falls above the mutation rate, it is assigned a value of 0. A different mask is generated for each child. Each mask is then applied to the child after the crossover. The Model takes the mask and overlays it onto the child. Where a mask value of 1 falls on a child, that cell is switched from 0 to 1 and 1 to 0. This simulates a mutation, but is a modification of the GA process. The mutation rate assigned by the user is not a de facto for the amount of mutation, rather it is a guideline. In some cases, a mutation may not occur, and when the mutation does occur, typically it is greater than the rate the user chose. Nevertheless, the mutations are allowed to occur and they take into consideration the Model constraints.

2.2.4.4 Constraints and Rules

To sidestep many of the limitations and problems associated with the model, a couple of constraints and rules needed to be established. These constraints would be considered heuristics or modifications of a pure GA approach. Therefore, a description of the constraints is given in this section. The first, and most important, is a contiguous constraint placed upon the harvest design. The constraint ensures that in any harvest plan the following must not occur:
Any black cells that are not touching must be in groups larger than three adjacent cells. In the case of the two cells in the top right corner, the contiguous constraint would eliminate these cells as potential harvest cells.

Any groups of cells that are three or more in length, without any other adjacent cells in between, should not exist.

Any cells that are in a diagonal trend should not exist because they do not have any touching neighbors as defined by the rule.

**Figure 2.5 Cases where constraint is violated**

Some examples of what the constraint does allow are:

- A stair-step or minimum grouping of three cells.
- Lines of cells of any length, as long as there is an adjacent neighbor every other cell.

**Figure 2.6 Cases where constraint is not violated**

These constraints help to simulate and encourage harvest plans that incorporate some operational level requirements. The code required to implement these rules took an incredible amount of time to build because there are so many different shapes, arrangements and orders in which these configurations could occur. Therefore, for the sake of time, the filtering mechanism for this constraint used a 3x3 cell analysis window. If the analysis window size was increased, the amount of artificial intelligence required to program could have increased exponentially. For example, in a 3x3 cell window, consider the number of arrangements which contain one cell with two or more adjacent cells of the same value and which do not occur in a straight line:
Then consider similar patterns within a 3x3 cell grid, but that also have additional attached cells. The number of potential contiguous possibilities increases dramatically. Naturally, one can see that increasing the filter size to a 5x5 matrix would result in a greater variety of potential arrangements and complexity. Therefore, all contiguous constraints were limited to exist within the range of one direct neighboring cell and no further. A minimum level of three adjacent cells was chosen by default because it is also the same pattern that allows for a stair-step growth away from the large patch. It was also used as a minimum patch size in an attempt to approach some level of operational integrity. Otherwise, a harvest plan could have looked like a patchwork quilt, having no applicability to operational planning. This constraint helps to bring some order to potential chaos.

The other rule applied to the model is a method which modifies a harvest plan by increasing the number of cells designated for harvest, such that it meets the harvest requirement. This growth algorithm uses the contiguous requirement to find all cells that are available for harvesting, then randomly selects a cell from those available cells using a random number generator and adds the cell to the harvest plan. Once the cell is added to the harvest, its neighboring cells are examined to see if they qualify as valid cells to harvest as well. Each neighboring cell that meets the contiguous requirement is added to the list of cells available for harvest.

2.2.4.5 Completion

The final step in the GA for this Model is the way in which completion, or the final solution, is determined. The completion of the Model can occur in one of two ways. Either the Allowable Visible Alteration (AVA) is met by one child solution or the Maximum Number of Iterations is
reached. The first possibility is fairly straightforward: it states that if the VQO is met by the harvest
design, then end the process. The fitness used to determine if the harvest design meets the VQO
is detailed in Appendix 9. The second completion possibility is a method that was defined
primarily by the limitations of ArcInfo, but also for greater control by the user. In the second
possibility, the Maximum Number of Iterations enables the user to specify a number of
generations to create so that potential application crashes to not occur. On some computers,
ArcInfo may crash after 20 iterations, while on more powerful computers the Model can run for 45
generations. Establishing a limitation on the number of times the Model runs ensures that fewer
crashed occur. Furthermore, allowing the Model to be executed for a limited number of iterations
provides an opportunity for the user to free up system resources when the computer is being
used for other applications. During a manual halt or crash a record of each generation’s solutions
are kept such that the model can be restarted at the last generation. If a solution is not found to
meet the AVA, then the user can choose to manually start the Model at the last generation.

2.2.5 Input

For the model to operate, a list of input requirements and parameters is necessary. The list is
necessary to fulfill a variety of different scenarios while assuring the user a high degree of
flexibility that allows for a tailored result of each problem. Each of these requirements is listed
below with a brief description of its characteristics and why it applies. The inputs are separated
into two different classes: data and variable inputs. The data variables are inputs such as raster
and vector files, while the variable inputs are objects like coordinate system, integers and strings.
The data inputs will be listed first, followed by the variable inputs.

2.2.5.1 Data Input

Ten initial data variable inputs are required by the model. Nine of these inputs are in raster format
and one is a shapefile input. In the examples given below, a simple 25x25 cell scenario was
created, with an extent covering 60x60 cells and a resolution of 60 meters. This was done to
ensure that the viewpoint, which would exist in the far upper right corner of the raster had an
elevation value associated with it. The extent was created as such because it allows for the
viewpoint to be simulated from middle ground distances, roughly 3km. Further details are given below:

**Name:** DEM (Digital Elevation Model)  
**File Type:** raster  
**Cell Size/Resolution:** 60x60, 60m  
**Description:**  
The DEM is used as the extent for the entire project. The most important piece about the DEM is that it has a valid value of 0 or more at the same location of the viewpoint; otherwise the analysis will not occur. In the case presented to the right, a valley is shown with white representing hilltops and black representing sea-level. The viewpoint will be positioned in the top right corner of the DEM.

*Figure 2.8 A Scenario DEM raster*

**Name:** targetCut (Target Cut)  
**File Type:** raster  
**Cell Size/Resolution:** 25x25, 60m  
**Description:**  
targetCut defines the potential harvest area, or area that the model is allowed to create a harvest plan within. The extent of this raster is its own bounds. The raster values consist of either zeros or ones. Any value of 1 designates a potential cell for harvesting. In the example shown here, a simple square is given, each cell with a value of 1.

*Figure 2.9 A Scenario Target Cut raster*

**Name:** ParentA  
**File Type:** raster  
**Cell Size/Resolution:** 25x25, 60m  
**Description:**  
ParentA is the first of two required inputs harvest plan inputs. A well defined harvest plan is not necessary, but it can provide the model with an initial plan from which to modify. Alternatively, the user may choose a random harvest plan. In some cases, this could greatly enhance the speed to which a solution is found. This raster also contains zeros and ones: ones representing the gray (or cut). The extent is also its bounds. ParentA is part of the initial analysis to see if the plan meets the fitness requirement.

*Figure 2.10 A Scenario Parent A raster*
Name: ParentB  
File Type: raster  
Cell Size/Resolution: 25x25, 60m  
Description:  
ParentB is the second of two required inputs harvest plan inputs. Although these do not have to be anything fancy, it does allow for the user to use a predetermined harvest plan, or intelligence, before running the model. In some cases, this could greatly enhance the speed to which a solution is found. This raster also contains zeros and ones; ones representing the gray (or cut). The extent is also its bounds. ParentB is part of the initial analysis to see if the plan meets the fitness requirement.

Figure 2.11 A Scenario Parent B raster

Name: treeHeight (Tree Height Map)  
File Type: raster  
Cell Size/Resolution: 60x60, 60m  
Description:  
treeHeight is an input that simulates a number of different natural and non-natural features such as hydrological features, roads, rocks or similar features, or an area of non analysis such as the black values shown. This raster is blanketed on top of the DEM, providing the basic visibility analysis raster. It also allows for a variety of tree heights or vegetation, consistent with the variety of vegetation heights on a forest landscape, to be input,. The extent of this raster is the DEM. The values consist of integers that represent the number of units (generally meters) above ground level which are specified as zeros.

Figure 2.12 A Scenario Tree Height Map raster

Name: VSU (Visual Sensitivity Unit)  
File Type: raster  
Cell Size/Resolution: 25x25, 60m  
Description:  
The VSU is the area of analysis that is used to measure all the visible surfaces that have negative visual impacts. These areas could include past and recent clear-cuts, roads or a variety of other features. The VSU is not limited to any particular shape but it should try to simulate a VSU within a 3D context. The extent of this raster is its bounds. It consists of ones and NoData; ones being locations of the VSU.

Figure 2.13 A Scenario Visual Sensitivity Unit raster
2.2.5.2 Variable Input

Thirteen variable inputs, three of which are optional, are required for the model to run properly.

Most of the input values are fairly intuitive, but some require further explanation. The variables can be split into two different designations, numerical and non-numerical values. Details about the numerical variables are presented first, followed by non-numerical variables:

2.2.5.2.1 Numerical Variable Input
### Table 2.2 Model's Numerical Input

<table>
<thead>
<tr>
<th>Variable Name</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Population Size</td>
<td>Integer</td>
<td>This value represents the number of children in each generation. The default value is 10.</td>
</tr>
<tr>
<td>Start Generation Number</td>
<td>Integer</td>
<td>This variable allows for the model to be documented properly by indexing the starting generation. So, if the model is run for a certain number of generations and then completes without finding a solution, the model can be initialized again, with this variable set to be the next generation in order.</td>
</tr>
<tr>
<td>Maximum Number of Iterations</td>
<td>Integer</td>
<td>This is the number of generations to run. There is a default of 99 generations, but this has never been achieved because ArcInfo crashes before it happens. This variable allows the user to run the model only for a few generations, perhaps while the computer is idle.</td>
</tr>
<tr>
<td>Size of Initial Population</td>
<td>Integer</td>
<td>This is the size of the initial population. Using the data inputs described in the previous section, this value would be 2, because ParentA and ParentB are the initial population.</td>
</tr>
<tr>
<td>Allowable Visible Alteration</td>
<td>Decimal</td>
<td>This variable determines the allowable visual impact requirement, or VQO, that can be assessed within the VSU. The value must fall between 0 and 100.</td>
</tr>
<tr>
<td>Required Timber Extraction</td>
<td>Decimal</td>
<td>This is the amount of required harvest, between 0 and 100 that the user requires for the harvest plan.</td>
</tr>
</tbody>
</table>

### 2.2.5.2.2 Non-Numerical Input

### Table 2.3 Model's Non-Numerical Input

<table>
<thead>
<tr>
<th>Variable Name</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analysis Resolution</td>
<td>Cell Size</td>
<td>Although this variable seems like it should be an integer value, ArcInfo requires this to be of type Cell Size. The user, however, is prompted for an integer value and should insert an integer value. ArcGIS will do the conversion to type Cell Size.</td>
</tr>
<tr>
<td>Coordinate System</td>
<td></td>
<td>This defines the projection and coordinate system that will be affixed to every created raster within the model.</td>
</tr>
<tr>
<td>Working Location</td>
<td>File</td>
<td>This should be specified as the initial modHrvtost data input. This is required by a script that was created manually and specifies the working location, or location to save the harvest plans in.</td>
</tr>
<tr>
<td>Temporary Processing Location</td>
<td>Folder Location</td>
<td>This is the temporary processing directory where all the parts, supplemental and supportive, raster and text files are created.</td>
</tr>
<tr>
<td>Scenario Name</td>
<td>String</td>
<td>This is primarily for documentation purposes and is recorded in the log file.</td>
</tr>
</tbody>
</table>
### Variable Name | Data Type | Description
---|---|---
Log File Name | File | This is the location of the text file to which the model will write all numerical results or visible alteration of each child for each generation.
Log File | Boolean | This is a checkbox of true/false that allows the user to choose whether or not to log the results numerically.

#### 2.2.6 Output

The output the model generates consists of two major components. The first and most important output is the final generation harvest plans, called modhrvstX, where X is the child number 0 to Population Size. The second output is the pseudo visibility analysis of each child's harvest plan, called visuglyX, where negative visual features are visible. These features will include the hiContrastSrf and the surface from the harvest plan and are measured over the entire extent of the VSU. These two outputs both consist of a raster format and each has the same resolution. The harvest plan raster will have the same cell size as the targetCut and the visibility analysis raster will have the same cells size as the VSU.

After the model has returned a successful solution, the user will be given the best solution as ParentA. This raster will be a copy of the best harvest plan, modhrvstX, as measured by visuglyX. Before the solution can be analyzed further or applied to the planning process, the user may want to convert it to an understandable format. The best solution, ParentA, consists of three different values: 0, 1 and 2. Non-zero values should be reclassified to a value of 1 to best represent the final solution. A few other additional outputs that the user will receive are copies of each generation's results so the user can track the changes over time. Finally, if the user opted for a log file to be written, then the record of each child's numerical representation of visuglyX will be documented.
2.3 The Model Apparatus

2.3.1 Description of Components

Within the scope of the Model, an array of processes, steps, tools and models are used. In fact, seven models, three scripts and dozens of ArcInfo tools are used to take input, iterate through 48 unique steps and create final output. The Model cannot be considered a pure Genetic Algorithm. Some alterations of the GA approach have been made in order to reduce the time it takes for the program to find a suitable harvest plan. The pseudo-GA approach does, however, follow the basic six step GA approach outlined at the beginning of this chapter.

The Model has been broken down into three steps for purposes of organization and as workarounds to some ArcInfo limitations. The first step in the Model does not comply to the standard GA approach. Rather, it is designed strictly as a processing time saver by helping to generate an initial design using local knowledge. The next two steps in the Model fulfill the six steps mentioned in the GA approach section. The second step in the Model operates steps 1-3 in the GA approach and step three in the Model operates steps 4-6 in the GA approach.

2.3.2 Steps

The Model is divided into the sections that will be referred to as Step #1, Step #2 and Step #3. Step #1 searches for all areas that will never be visible to the view point. Step #2 takes the findings from Step #1 and combines it with the user input harvest plans shown in Figure 2.10 and Figure 2.11. Then it verifies if any of the harvest plans meets the requirements (VQO and timber extraction levels). If the neither of the plans meet the requirements then Step #3 is ran. Step #3 is the iterative process that takes the harvest plans from Step #2, breeds them, assesses the output and continues until a solution is found or the user halts the continuation of the program.

2.3.2.1 Step #1

This step, effectively an ArcInfo model, consists of ten processes and one script that finds any region which is invisible from the view point. It then allocates these areas as definite harvest areas and eliminates them from the targetCut designation. Doing this decreases processing time
and ensures the greatest opportunity for reduced visual impact. Each process is explained in a more detailed format in Appendix 4.1. For a diagram of this step, please refer to Appendix 3.1.

2.3.2.2 Step #2
As mentioned earlier, this step fulfills the first three steps in the GA approach. Again, the first three steps in the GA approach are: (1) select initial population of solutions, (2) determine the fitness for each solution, and (3) choose the best solutions based on fitness. In the case of the model presented, the initial population is set as only two parents, so there is no step that determines the best solutions because the two best already exists. However, the model for this step does order the output solutions as ParentA and ParentB, the best and second best solution, respectively. This model contains six processes, three of which are models themselves. A brief description of these three models will be given, but details will be left out because they are duplicated and explained in full, in step #3. After running this model, the Best Percent Visible Alteration is returned. This value can be matched against the Allowable Visible Alteration (AVA) to verify if one of the input solutions meets the AVA requirement. If one of the solutions does meet the AVA, then the Model is complete and ParentA is the final harvest plan solution. Each process is explained in a more detailed format in Appendix 4.2. For a diagram of this step, please refer to Appendix 3.2.

2.3.2.3 Step #3
The most complex and detailed model, Step #3, includes 4 models, combining a total of 34 processes. This step uses an iterative procedure loop that continues searching for a solution until it meets the Allowable Visible Alteration requirement or exceed the maximum number of iterations. This step fulfills the last three steps in the GA approach, which are: (1) Breed the new generation through crossover of parents and mutation, (2) Determine the fitness for each new solution, and (3) Loop to choose the best solutions based on fitness or terminate if a solution is found. One deviation from the GA approach happens on the last step of this model. The deviation is a variable that allows the user to escape from the iterative process. This was included in the model because ArclInfo has some significant memory leaks that limit the number of loops without
crashing. Two variables were introduced that allow the model to stop after a certain number of iterations (Maximum Number of Iterations) and restart, beginning with the last iteration (Start Generation Number). The terms iteration and generation are synonymous with one another because one loop ultimately reflects another generation of children. After each loop, the model returns the Best Percent Visible Alteration from the generation’s children. This value is then matched against the Allowable Visible Alteration (AVA) to verify if one of the input solutions meets or exceeds the AVA requirement. If one of the solutions does meet or exceed the AVA, then the Model is complete and ParentA is the final harvest plan solution. If one of the solutions does not meet the AVA and the Maximum Number of Iterations is not yet met, then another loop begins. Each process is explained in a more details format in Appendix 4.3. For a diagram of this step, please refer to Appendix 3.3.

2.4 Measuring Perspective View

As discussed in the previous sections, the Model uses a pseudo-perspective visible alteration algorithm. However, an alternative algorithm was developed to enhance the precision and accuracy with which the perspective visible alteration can be measured. The former method uses a pixel count divided by the number of pixels in the VSU as its basis for percent. So, if a VSU contains 625 cells, as is the standard VSU for all hypothetical scenarios used in the study (see Section 2.5.1), and the number of visible cells is with a high contrast surface is 25, then the percent visible alteration (PVA) is 25/625 or 4%. To derive the number of cells visible in perspective view for the former algorithm, the viewshed analysis tool in ArcInfo is used.

The viewshed analysis tool uses an altered terrain that is a combination of the DEM and Tree Height Map to generate visible cells. From this terrain, a cut is simulated by extracting those harvest areas by subtracting the height of the trees within the harvest from the altered terrain. The final outputs of these two steps are a terrain map that includes the height of the trees and the full harvest of cells. Then the viewshed analysis using the viewpoints.shp and the terrain raster is run. This generates a raster that separates the visible (value of 1) and invisible (value of 0) cells. The tool determines visible from invisible cells by comparing the altitude angle of the center of
each cell with the altitude angle to the local horizon (which is the viewpoint). Details of how this method determines if a cell is visible or not can be found in Appendix 8. The result

In the latter method a greater amount of precision is available by considering a weighted function of the amount of light that a cell reflects back to the viewpoint. This method was not included in the Model, but was developed after and as a supplement to the existing processes. The intent was to find a more precise calculation method that would conduct a more realistic assessment of the percent visible alteration. Then, apply the method to the results of the Model in order to see if any differences between the two calculation methods exist. Furthermore, integrating this method as a supplement shows how a weighted raster can be adapted into the current fitness value to arrive at a solution. This method was done by combining the hillshade tool available in ArcInfo and the viewshed analysis tool together. For details on how this is done, refer to Appendix 9.

2.5 Scenarios

Several scenarios were created in order to test the functionality, applicability and adaptability of the Model to find a harvest design with the lowest visual impact. Using an array of spatial and non-spatial parameters, each scenario was carefully crafted so that the variety of solutions and differentiation between solutions could be easily seen. These scenarios were also created in order to simulate potential landscape scenarios, but some were simplified to make testing and comparisons easier. All scenarios were tested using an AVA of 1% because it was thought that this would challenge the model's capabilities to find low visual impact solutions. It could have been just as easy to apply any percentage level, such as 20% or 0% AVA, but 1% was chosen to both test the Model's capabilities and produce plans that could be solved in a reasonable amount of time. Using a 1% level, rather than a 7% level means that the solution has the potential to have that much less of an impact. In fact, the Model keeps a record of all past solutions, so if a certain percentage is desired, say 8% all solutions that fall between 8% and less are optional harvest solutions. Seven hypothetical scenarios were generated from scratch and two scenarios derived from real-world data were also created. Two different dimensions, viewpoint height and Required Timber Extraction Levels, were modified to create an array of scenarios.
2.5.1 Hypothetical Scenarios

All hypothetical scenarios were run at the same resolution and cell size, using two different viewpoint heights. The same Target Cut and VSU raster files were used across hypothetical solutions, in order to streamline testing and ensure consistency between scenarios. A cell size of 60 meters was used because this would be the minimum feasible size block operations, using ground equipment, could occur (Nelson 2007). When the Target Cut map used for these scenarios was converted to hectares by multiplying the number of cells (25 * 25) times the cell resolution (60 * 60m), this resulted in a potential harvest size of 225 hectares.

In regards to the variable input, the following were used for each scenario:

<table>
<thead>
<tr>
<th>Variable</th>
<th>Input Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Population Size</td>
<td>10</td>
</tr>
<tr>
<td>Start Generation Number</td>
<td>1 (this number changed if Step #3 needed to be run again)</td>
</tr>
<tr>
<td>Maximum Number of Iterations</td>
<td>30 (This was usually set to thirty, but depended upon conflict with other software using the computer's resources)</td>
</tr>
<tr>
<td>Size of Initial Population</td>
<td>2</td>
</tr>
<tr>
<td>Analysis Resolution</td>
<td>60 meter</td>
</tr>
<tr>
<td>Coordinate System</td>
<td>NAD_1983_UTM_Zone_11N</td>
</tr>
<tr>
<td>Working Location</td>
<td>&lt;model directory&gt;&lt;Data&gt;modhrvst0</td>
</tr>
<tr>
<td>Temporary Processing Location</td>
<td>model directory&gt;temporary</td>
</tr>
<tr>
<td>Log File name</td>
<td>&lt;model directory&gt;&lt;scripts_files&gt;log.txt</td>
</tr>
<tr>
<td>Log File</td>
<td>True</td>
</tr>
</tbody>
</table>

The remaining variables were changed depending upon the scenario:

1. Scenario Name
2. Allowable Visible Alteration
3. Required Timber Extraction

The remaining data presented in this section provide detail about the input parameters used to establish each scenario. All data input are the same as the example shown in the Data Input section, with the exception of the DEM. Therefore, the DEM has been provided as well as the other parameter details. The range of the visibility analysis was roughly 3km as measured from the viewpoint to the front of the VSU, where the change in landscape features begin. From the viewpoint to the front of the VSU, the ground is flat to ensure visibility to the front of the VSU.
the front of the VSU, the terrain changes to reflect the DEM and scenario parameters. All the hypothetical DEMs that were generated were verified for slope to ensure that each pixel does not have a slope percent greater than 60%, the maximum slope the BCMoF allows for harvest. All tree heights are set to 20 meters to simulate stands ready for harvest. The Parents were not created with any sort of purpose, but rather as simple beginning points to emphasize the Model's ability to redraw and modify the harvest to fit each scenario.
2.5.1.1 Scenario #1: Flat

Dimensions Tested

<table>
<thead>
<tr>
<th>Allowable Visible Alteration (%)</th>
<th>Required Timber Extraction Levels (%)</th>
<th>View Height</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 (Retention VQO) 80%</td>
<td></td>
<td>1.5m</td>
</tr>
<tr>
<td></td>
<td></td>
<td>41.5m</td>
</tr>
</tbody>
</table>

This scenario was intended to simulate a flat and level surface over the entire landscape. The objective was to test the speed and search algorithm of the system to find a harvest plan that met the requirements. In a flat landscape, the solution should leave a barrier of trees between the viewpoint and the harvest. This scenario was built as the primary and simplest test of the Model's capabilities.

2.5.1.2 Scenario #2: Four Medium Hills

Dimensions Tested

<table>
<thead>
<tr>
<th>Allowable Visible Alteration (%)</th>
<th>Required Timber Extraction Levels (%)</th>
<th>View Height</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 (Retention VQO) 80%</td>
<td></td>
<td>1.5m</td>
</tr>
<tr>
<td></td>
<td></td>
<td>41.5m</td>
</tr>
</tbody>
</table>

This scenario was intended to simulate a surface with four small hills at a maximum height of 40 meters. The intention was to test the Model's ability to use the hills to hide the cut behind intact trees.

2.5.1.3 Scenario #3: High Mountain Face

Dimensions Tested

<table>
<thead>
<tr>
<th>Allowable Visible Alteration (%)</th>
<th>Required Timber Extraction Levels (%)</th>
<th>View Height</th>
<th>Resolution</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 (Retention VQO) 60%</td>
<td></td>
<td>1.5m</td>
<td>60m</td>
</tr>
<tr>
<td></td>
<td></td>
<td>41.5m</td>
<td>20m</td>
</tr>
</tbody>
</table>

This scenario was intended to test the Model's algorithm against high sloped landscapes. This DEM creates a high slope mountain with a peak height of 400 meters, which allows for the entire slope to be visible from the viewpoint. There are only a few small areas that are initially not visible because they have the same height as the surrounding area and are not visible because the curvature of the earth hides them. Slopes range from 0%-60%, with the 0% slope at the top and right edges of the DEM. This scenario is the best test of the screening capabilities because of the extremely high slopes, which limit the capacity for screening. Two different resolutions were tested to see if there would be any effect for a landscape with high slopes.
2.5.1.4 Scenario #4: One High Hill

<table>
<thead>
<tr>
<th>Dimensions Tested</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Allowable Visible Alteration (%)</strong></td>
</tr>
<tr>
<td>-------------------------------------------------------</td>
</tr>
<tr>
<td>1 (Retention VQO)</td>
</tr>
</tbody>
</table>

This scenario was intended to test the Model's viewshed ability to deal with three dimension curvature on a hill. The "wrapping" effect of the hill has the potential to hide other cells behind it. This scenario was also used to test the Model's ability to hide a cut, as analysis for this DEM is easy to see with the naked eye. The highest point on the hill is 75 meters.

![Figure 2.19 Scenario #4 DEM](image)

2.5.1.5 Scenario #5: One High Hill and Three Medium Hills

<table>
<thead>
<tr>
<th>Dimensions Tested</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Allowable Visible Alteration (%)</strong></td>
</tr>
<tr>
<td>-------------------------------------------------------</td>
</tr>
<tr>
<td>1 (Retention VQO)</td>
</tr>
</tbody>
</table>

This scenario was intended to simulate a rolling hill surface, where only part of the hill in the background is visible. This allows for the testing of a harvest plan that considers multiple levels of surfaces into its final plan. If a harvest plan cuts the areas in the back hill, while the areas in the foreground hill block the background, then that satisfies the requirement. However, if in a subsequent generation the foreground hill becomes part of the harvest plan, then the penalty should be greater because now both the surface from the foreground and the surface form the background become visible. The peaks in the small hills are 15 meters and the peak in the background hill is 80 meters.

![Figure 2.20 Scenario #5 DEM](image)

2.5.1.6 Scenario #6: Two High Hills

46
### Dimensions Tested

<table>
<thead>
<tr>
<th>Allowable Visible Alteration (%)</th>
<th>Required Timber Extraction Levels (%)</th>
<th>View Height</th>
</tr>
</thead>
<tbody>
<tr>
<td>40%</td>
<td></td>
<td>1.5m</td>
</tr>
<tr>
<td>45%</td>
<td></td>
<td>41.5m</td>
</tr>
<tr>
<td>50%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>55%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>60%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>65%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>70%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>75%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>80%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>85%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>90%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>95%</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

This scenario was intended to simulate a very simple valley problem and the Model’s ability to hide the harvest in the background. The high points on these hills are 70 meters. This scenario was created as a sensitivity test which required different timber extraction levels. Pilot tests showed that this scenario required a significant number of generations to be run before nearing a “best-fit” scenario. Therefore, this scenario was established because it offers both ease of analysis, because of the simplicity of the terrain, and a terrain complex enough to require unique design.
2.5.1.7 Scenario #7: Two High Mountains with a Valley

<table>
<thead>
<tr>
<th>Dimensions Tested</th>
</tr>
</thead>
<tbody>
<tr>
<td>Allowable Visible Alteration (%)</td>
</tr>
<tr>
<td>1 (Retention VQO)</td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

A more complex scenario representing a valley was created to test the Model’s viewshed ability to find those areas that are invisible, but contiguous with other parts of a harvest plan. This valley has several dips and turns that draw it closer to a realistic valley corridor. Several peaks and high points range from 80-200 meters, with lows between 0-40meters. A three-dimension representation of this DEM is given below to emphasize the complexity of the DEM. Figure 2.23 is a 3D representation. The bottom of the DEM is the same as the top right corner of the DEM in Figure 2.22.

2.5.2 Realistic Scenarios

2.5.2.1 Tree Farm License 49 Near Kelowna, BC

This scenario was created as a proof of concept to show that the model can be used in more complex situations than the hypothetical scenarios presented earlier. This scenario captures the way which the Model uses a variety of different inputs, including a VSU, Tree Height Map and High Contrast Surface raster files to produce output that is constrained to those input requirements. The Target Cut raster is 96x111 cells at a 25m resolution, far larger than the hypothetical scenarios. The distance from the view point to the center of the harvest area is roughly 5.5Km. The area of this scenario was built on forest cover data and TRIM DEM data. The
location is near Kelowna, BC. In a WGS 1984 projection, the exact location of the viewpoint is 119°26'3.18"W and 50°11'24.064"N. The site chosen for harvest was selected because it is one of the few areas of the TFL that is visible from the east side, where the Okanagan Lake and city are located. The specific site is unknown in terms of its usage or popularity, but it is accessible by road.

This is the DEM that is used in this scenario. A manual modification of the DEM was made in order to correct for areas that did not have any forest cover information. In these cases, cells were given a 10m increase. Where cells had forest cover information, they were left as the original height in the DEM. From the bottom left to the top right is about 6km in length. The viewpoint will be located in the bottom right of the DEM. The area of harvest will be around and between the two white areas of the DEM. The DEM and all subsequent raster files are 25m resolution. The viewpoint is located in the bottom right corner.

The tree height map was derived from the forest cover data. First the average tree height between the maximum and minimum tree heights within each section in the TFL were derived. Then, a conversion from the raw shapefile to raster format using the average height values was produced. This formed the basis for the tree height map.
The VSU was produced by designing a rough sketch of a potential VSU. The sketch was not meant to fit process whereby the Forest Manager develops a VSU, rather it was to demonstrate the models ability to solve a realistic problem. If the VSU information was available, that would have been used. Then all cells within the forest cover that had a slope greater than 60% were eliminated from the VSU, since that is generally considered the maximum slope for logging.

Figure 2.26 VSU for TFL 49

The high contrast surface was generated from the forest cover raster. All cells in the forest cover raster that fell below 7.25m were set to a high contrast surface. This was done *ad hoc* and does not reflect a specific process. Differentiating these heights was reasonable as they suggest recent cuts. Again, if time and information was provided, a more suitable map could have been developed and used within this model. The gray cells are those that do not have a high contrast surface. The Black cells are those that are recent clear-cuts or human-built structures.

Figure 2.27 High Contrast Surface for TFL 49
The target cut was created by extracting the high contrast surfaces from the VSU. This consists of a 96x100, 25m cell resolution grid.

Figure 2.28 Target Cut for TFL 49
3 Results

This section of the thesis details several different results. The primary focus is on the outcome from each dimension of each scenario that is identified in the Methods section of this document. The secondary focus is on a comparison between the effects of changing the timber requirement dimension in one (Picard and Sheppard 2001a) showing the relationship between VQO intensity and theoretical timber availability. Other statistics and results about the overall outcome of the model will be given for supporting evidence, context and critiquing purposes.

3.1 General Outcomes

Results have been compiled from 28 different hypothetical scenarios and one realistic scenario ranging from 20 x 20 cell grids to 96 x 111 cell grids, using 20, 25 and 60 meter resolution cells. The number of generations required to find a solution ranged from 0 to 451 generations. In some cases no generations were required to complete the model as the initial parents sufficed and therefore no GA was used to find additional solutions. Some scenarios, including the one which required 451 generations, were stopped short of the desired allowable visible alteration (AVA). If a process was stopped before the program found a solution to meet both the AVA and the required timber extraction levels, it was because no improvement had been made after several generations.

A number of different dimensions were tested, including viewing height, required timber extraction levels and spatial resolution. The effect of changing the required AVA was not completed because the specified level, 1%, was deemed a very low level and thus a reasonable cut-off point. Also, the effect of altering the AVA was not tested, because preliminary tests suggested that many scenarios would have been completed in very few generations. This suggests for practical intent, that solutions where the AVA is higher than 1% may be found quickly and in few generations. Essentially, by trying to establish an AVA of 1% for these hypothetical and simplistic scenarios tested the GA.
The computing time necessary to complete each scenario was painstakingly slow. It appears that when using ArcInfo's Modelbuilder, ArcInfo requires a lengthy processing time. When Modelbuilder references a tool in ArcInfo, the system spends a few seconds for each run of each tool. Several different computers ran the model. It was discovered that the computers which ranged from a 2.5Ghz-3Ghz AMD® or Pentium 4® processor with a range of 1-3 GB of RAM took approximately 10 minutes to run through one iteration. So in the scenario that took 451 generations to run this totaled to be about 75 hours and took roughly 4 days to run. The older machines which had a 1GHZ Pentium 3® processor and 0.5 GB of RAM took approximately 15 minutes to run through one generation. Processing time was not subject to change between the lower and higher resolution raster files, meaning that it was likely the tools within ArcInfo that caused the delay. Tests were not run to determine what computational characteristic(s) affected the speed of processing. The efficiency and run-time was an expensive cost and did not afford much time for sensitivity analysis.

3.2 Scenario Outcomes

In the following section data from a few key scenarios will be reported. A table format will be used in order to provide consistency between datasets for easy comparison. A brief explanation of what data will be contained is given in Table 3.1. Depending upon the scenario, certain categories will not be presented. For a detailed report of all scenarios refer to Hypothetical Scenarios Appendix 6. Appendix 6 reports all data for every category in Table 3.1.
Table 3.1 Table structure used to report scenario results

<table>
<thead>
<tr>
<th>Required Timber Harvest Extraction Level</th>
<th>The percent of required timber harvest</th>
</tr>
</thead>
<tbody>
<tr>
<td>Desired AVA</td>
<td>The desired allowable visible alteration (AVA), or VQO, that might be established. Note that each of the scenarios has an AVA of 1%. Setting a value this low, which falls into the retention VQO, forces model to run until that level is met. In many cases, a solution was found with the required AVA, but some scenarios had to be manually ended because a better solution could not be found that met the desired AVA.</td>
</tr>
<tr>
<td>Viewing Height</td>
<td>This is the height of the viewpoint, simulating a person standing at the lowest level in the scenario DEM or raised to simulate standing on top of a hill.</td>
</tr>
<tr>
<td>Invisible Cells</td>
<td>An image of the viewshed analysis in Step #1. The gray in the image determines the invisible cells, and ensures that all those cells are set for harvest. This shows both the cells that are invisible and the cells that are visible in order to depict the initial amount of visible cells for the DEM.</td>
</tr>
<tr>
<td>Initial Parent AVA</td>
<td>The percent visible alteration values of each of the initial parents after the completion of step #1, which takes an initial harvest plan, identifies areas invisible and ensures that the required amount of harvest is selected.</td>
</tr>
<tr>
<td>Number of generations until completion</td>
<td>The number of generations (or iterations) the model ran to arrive at the final solution</td>
</tr>
<tr>
<td>Type of completion</td>
<td>Automatic means the algorithm found a solution that met the desired AVA. Manual means the Model's user had to stop the model because too many generations were run that did not result in any better solution.</td>
</tr>
<tr>
<td>Best Percent Visible Alteration found</td>
<td>The percent visible alteration from the best solution.</td>
</tr>
<tr>
<td>Best harvest solution found</td>
<td>An image of the best harvest solution, gray is the area designated for harvest. The viewpoint is located beyond the top right corner of the image looking toward the bottom left corner.</td>
</tr>
<tr>
<td>Visible cells after harvest of best solution</td>
<td>An image of the best harvest solutions' visible spots on the surface with a negative impact. The gray cells are those cells which are visible. The viewpoint is located beyond the top right corner of the image looking toward the bottom left corner.</td>
</tr>
<tr>
<td>Output log data</td>
<td>A graph showing the progression from first to last generation after the initial selection process. For every generation (Y axis) the worst and best percent visible alteration is reported (X axis), as well as the average percent visible alteration of all 10 children. A trendline and slope values depicting the trend of the best (lowest) solution over time is also displayed. The trendline was generated using a logarithmic function to depict a leveling off toward the best possible solution. This data records the highest, average and lowest Best Percent Visible Alteration for every generation and records that data on the Y axis.</td>
</tr>
</tbody>
</table>
Simple visualizations of a few example scenarios have been created to demonstrate the final outcome in 3D format. Two images have been rendered, one from the view point at a height of 41.5m and one from a height of 1241.5m which gives a better representation of where the cut exists. These images were only intended to act as representations of a hypothetical area. The tree heights are set according to the tree height map at 20m high. The grey regions in the map that fall behind the front end of the VSU are those areas that were set for harvest. The gray areas in front of the tree front are filler values and were not included in the design phase, but were used to establish a view point. The images were rendered using Visual Nature Studio. The trees that are used to simulate a forest are images from the pine family and have a density of 250 trees per hectare. The importance was not placed on the species or the density, but a reasonable forest cover needed to be developed to show the final solution for this scenario in a simulated 3D environment.
3.2.1 Hypothetical Scenarios

3.2.1.1 Scenario #1: Flat

| Required Timber Harvest Extraction Level | 80% |
| Desired AVA | 1% |
| Viewing Height | 1.5m |
| Number of generations until completion | 17 |
| Best Percent Visible Alteration found | .48% |
| Best harvest solution found | |

![Figure 3.1 Scenario #1 1.5m Best Solution](image1)

The circle highlights the screening effect in the corner nearest to the viewpoint. This shows that the Model produces a screening effect as most all trees behind the front of the VSU cannot be seen.

| Required Timber Harvest Extraction Level | 80% |
| Desired AVA | 1% |
| Viewing Height | 41.5m |
| Best Percent Visible Alteration found | .8% |
| Best harvest solution found | |

![Figure 3.2 Scenario #1 41.5m Best Solution](image2)
### 3.2.1.2 Scenario #2: Four Medium Hills

<table>
<thead>
<tr>
<th>Required Timber Harvest Extraction Level</th>
<th>80%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Desired AVA</td>
<td>1%</td>
</tr>
<tr>
<td>Viewing Height</td>
<td>1.5m</td>
</tr>
</tbody>
</table>

#### Invisible Cells

![Figure 3.3 Scenario #2 1.5m Vis Invisible Cells](image)

A majority of the cells are classified as invisible because the three front hills (see Figure 2.17 for DEM) block much of the area shown as invisible cells.

<table>
<thead>
<tr>
<th>Initial Parent AVA</th>
</tr>
</thead>
<tbody>
<tr>
<td>ParentA: 2.88%</td>
</tr>
<tr>
<td>ParentB: 1.28%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Number of generations until completion</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Best Percent Visible Alteration found</th>
</tr>
</thead>
<tbody>
<tr>
<td>.96%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Best harvest solution found</th>
</tr>
</thead>
</table>

![Figure 3.4 Scenario #2 1.5m Best Solution](image)
<table>
<thead>
<tr>
<th>Required Timber Harvest Extraction Level</th>
<th>80%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Desired AVA</td>
<td>1%</td>
</tr>
<tr>
<td>Viewing Height</td>
<td>41.5m</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Invisible Cells</th>
<th><img src="image1.png" alt="Image" /></th>
</tr>
</thead>
</table>

**Figure 3.5 Scenario #2 41.5m Vis Invisible Cells**

<table>
<thead>
<tr>
<th>Initial Parent AVA</th>
<th>ParentA: 7.52%</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ParentB: 4.32%</td>
</tr>
</tbody>
</table>

| Number of generations until completion | 22 |
| Best Percent Visible                  | .96% |
| Alteration found                      |     |

<table>
<thead>
<tr>
<th>Best harvest solution found</th>
<th><img src="image2.png" alt="Image" /></th>
</tr>
</thead>
</table>

**Figure 3.6 Scenario #2 41.5m Best Solution**

The circle shows the rough area where a hill is. It highlights the area where the Model produced a screening effect on a hill face.
3.2.1.3 Scenario #3: High Mountain Face

Table 3.2 Scenario #3 comparison of cell size and timber extraction levels

<table>
<thead>
<tr>
<th>Scenario</th>
<th>#3a</th>
<th>#3b</th>
<th>#3c</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resolution</td>
<td>25 x25, 60m cell size</td>
<td>25 x25, 60m cell size</td>
<td>75 x75, 20m cell size</td>
</tr>
<tr>
<td>Required</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Timber Harvest</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Extraction Level</td>
<td>80%</td>
<td>60%</td>
<td>80%</td>
</tr>
<tr>
<td>Desired</td>
<td></td>
<td>1%</td>
<td></td>
</tr>
<tr>
<td>AVA</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Viewing Height</td>
<td></td>
<td>41.5m</td>
<td></td>
</tr>
<tr>
<td>Number of</td>
<td>195</td>
<td>170</td>
<td>146</td>
</tr>
<tr>
<td>generations</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>until completion</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Best Percent</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Visible</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Alteration</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>found</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Best harvest</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>solution</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>solution found</td>
<td>Figure 3.7 Scenario #3 41.5m Best Solution</td>
<td>Figure 3.8 Scenario #3 41.5m 60% Timber Best Solution</td>
<td>Figure 3.9 Scenario #3 41.5m 20m Res Best Solution</td>
</tr>
</tbody>
</table>

Figure 3.7, Figure 3.10, Figure 3.11 and Figure 3.12 are three dimensional renders of the two dimensional plans. The renders were created in Visual Nature Studio. All images were produced using the viewpoint at 41.5 meters. Figure 3.7 is the 2D plan for the 3D representation shown in Figure 3.10. Figure 3.8 is the 2D plan for the 3D representation shown in Figure 3.11. Figure 3.9 is the 2D plan for the 3D representation shown in Figure 3.12.
Figure 3.10 High Mountain Face 3D render at 41.5m viewpoint, 80% harvest

Figure 3.11 High Mountain Face 3D render at 41.5m viewpoint, 60% harvest

Figure 3.12 High Mountain Face 3D render at 41.5m viewpoint, 80% harvest, 20m cell
### Scenario #4: One High Hill

<table>
<thead>
<tr>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Required Timber Harvest Extraction Level</td>
<td>80%</td>
</tr>
<tr>
<td>Desired AVA</td>
<td>1%</td>
</tr>
<tr>
<td>Viewing Height</td>
<td>1.5m</td>
</tr>
<tr>
<td>Number of generations until completion</td>
<td>21</td>
</tr>
<tr>
<td>Type of completion</td>
<td>Automatic</td>
</tr>
<tr>
<td>Best Percent Visible Alteration found</td>
<td>.64%</td>
</tr>
<tr>
<td>Best harvest solution found</td>
<td></td>
</tr>
</tbody>
</table>

**Figure 3.13 Scenario #4 1.5m Best Solution**

<table>
<thead>
<tr>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Required Timber Harvest Extraction Level</td>
<td>80%</td>
</tr>
<tr>
<td>Desired AVA</td>
<td>1%</td>
</tr>
<tr>
<td>Viewing Height</td>
<td>41.5m</td>
</tr>
<tr>
<td>Number of generations until completion</td>
<td>239</td>
</tr>
<tr>
<td>Type of completion</td>
<td>Manual</td>
</tr>
<tr>
<td>Best Percent Visible Alteration found</td>
<td>1.44%</td>
</tr>
</tbody>
</table>

61
Scenario #4, 80% Extraction Rate
Log File Output

The following chart depicts the worst, best and average solution for the ten children in each generation (X axis). The solutions are measured in terms of the number of visible cells with a surface that is deemed high contrast. This is the visible alteration level. The logarithmic curve is displayed as a guide to show the leveling off effect of the solutions toward the lowest percent of visible alteration. In this particular scenario, the model required 239 generations, or iterations, until the program was manually stopped. The final result was a solution with a 1.44% visible alteration.
3.2.1.5 Scenario #5: One High Hill and Three Medium Hills

| Required Timber Harvest Extraction Level | 80% |
| Desired AVA                             | 1%  |
| Viewing Height                          | 1.5m|
| Initial Parent AVA                      | ParentA: 0.16%  
                                      | ParentB: 5.44% |

Invisible Cells

![Image of Invisible Cells](image)

Figure 3.16 Scenario #5 1.5m Vis Invisible Cells

<p>| Number of generations until completion | 0 |
| Best Percent Visible Alteration found | 0.16% |</p>
<table>
<thead>
<tr>
<th>Best harvest solution found</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Figure 3.17 Scenario #5 1.5m Best Solution" /></td>
</tr>
</tbody>
</table>

| Required Timber Harvest Extraction Level | 80% |
| Desired AVA | 1% |
| Viewing Height | 41.5m |
| Initial Parent AVA | ParentA: 2.24%  
ParentB: 6.24% |
| Invisible Cells |
| ![Figure 3.18 Scenario #5 41.5m Vis Invisible Cells](image) |
| Number of generations until completion | 9 |
| Type of completion | Automatic |
| Best Percent Visible Alteration found | .96% |
Best harvest solution found

Figure 3.19 Scenario #5 41.5m Best Solution

Figure 3.20 and Figure 3.21 are three dimensional renders of the two dimensional plan. The renders were created in Visual Nature Studio. The first image depicts the view from 41.5 meters to give a perspective view. The Second image depicts a different perspective view that lies between the plan view and the view point. This image more clearly emphasizes the screening effect and the clump of trees left on the base of the single high hill since that surface could be spotted from the view point at 41.5m
### 3.2.1.6 Scenario #6: Two High Hills

#### Table 3.3 Scenario #6 Comparison of timber extraction levels

<table>
<thead>
<tr>
<th>Scenario #</th>
<th>6a</th>
<th>6b</th>
<th>6c</th>
<th>6d</th>
<th>6e</th>
<th>6f</th>
<th>6g</th>
<th>6h</th>
<th>6i</th>
<th>6j</th>
<th>6k</th>
<th>6l</th>
</tr>
</thead>
<tbody>
<tr>
<td>Extraction Level (%)</td>
<td>95</td>
<td>90</td>
<td>85</td>
<td>80</td>
<td>75</td>
<td>70</td>
<td>65</td>
<td>60</td>
<td>55</td>
<td>50</td>
<td>45</td>
<td>40</td>
</tr>
<tr>
<td>Desired AVA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1%</td>
<td></td>
</tr>
<tr>
<td>View Height</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>41.5m</td>
<td></td>
</tr>
<tr>
<td>Number of Generations until Completion</td>
<td>451</td>
<td>280</td>
<td>236</td>
<td>245</td>
<td>143</td>
<td>53</td>
<td>24</td>
<td>13</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Best Percent Visible Alteration Found (%)</td>
<td>17.92</td>
<td>7.84</td>
<td>3.36</td>
<td>1.92</td>
<td>.96</td>
<td>.96</td>
<td>.96</td>
<td>.64</td>
<td>.64</td>
<td>.64</td>
<td>.16</td>
<td>0</td>
</tr>
<tr>
<td>VQO Type</td>
<td>Modification VQO</td>
<td>Preservation VQO</td>
<td>Retention VQO</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- provides a semantic alternative to percent visible altered.
### 3.2.1.7 Scenario #7: Two High Mountains with a Valley

<table>
<thead>
<tr>
<th>Required Timber Harvest Extraction Level</th>
<th>80%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Desired AVA</td>
<td>1%</td>
</tr>
<tr>
<td>Viewing Height</td>
<td>41.5m</td>
</tr>
<tr>
<td>Initial Parent AVA</td>
<td>ParentA: 4.16%</td>
</tr>
<tr>
<td></td>
<td>ParentB: 3.84%</td>
</tr>
<tr>
<td>Invisible Cells</td>
<td><img src="image" alt="Invisible Cells" /></td>
</tr>
<tr>
<td>Number of generations until completion</td>
<td>18</td>
</tr>
<tr>
<td>Type of completion</td>
<td>Automatic</td>
</tr>
<tr>
<td>Best Percent Visible Alteration found</td>
<td>.96%</td>
</tr>
<tr>
<td>Best harvest solution found</td>
<td><img src="image" alt="Best Solution" /></td>
</tr>
</tbody>
</table>

*Figure 3.22 Scenario #7 41.5m Vis Invisible Cells*  
*Figure 3.23 Scenario #7 41.5m Best Solution*
Scenario #7, 80% Extraction

Figure 3.24 and Figure 3.25 are three dimensional renders of the two dimensional plan. The renders were created in Visual Nature Studio. The first image depicts the view from 41.5 meters to give a perspective view. The second image depicts a different perspective view that lies between the plan view and the viewpoint. This image was rendered to emphasize the complexity of the landscape and show how the Model determined placement for harvested and retention areas.

Figure 3.24 High Mountain Valley 3D render at 41.5m viewpoint

Figure 3.25 Two High Mountain Valley 3D render at 1241.5m viewpoint
### 3.2.2 Realistic Scenario

#### 3.2.2.1 Tree Farm License 49, Kelowna, BC

Please note that the results from this scenario are continued over three pages.

<table>
<thead>
<tr>
<th>Required Timber Harvest Extraction Level</th>
<th>80%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Desired AVA</td>
<td>1%</td>
</tr>
<tr>
<td>Viewing Height</td>
<td>41.5m</td>
</tr>
</tbody>
</table>
| Initial Parent AVA                       | ParentA: 7.64%  
|                                          | ParentB: 7.62%  |
| Invisible Cells                          |     |

**Figure 3.26 Scenario #7 41.5m Vis Invisible Cells**

<table>
<thead>
<tr>
<th>Number of generations until completion</th>
<th>119</th>
</tr>
</thead>
<tbody>
<tr>
<td>Best Percent Visible Alteration found</td>
<td>6.77%</td>
</tr>
</tbody>
</table>
3.3 Effect of Percent Visible Alteration on Timber Availability

This section explains the effect of altering the timber extraction requirement while trying to find a harvest plan nearest to 1% AVA, or a VQO retention level. Scenario #6, Two High Hills, and the hypothesis developed by Picard and Sheppard (2001b) will be used to provide results that help explain the relationship between timber availability and VQO levels. The results from Scenario #6 will be used to show an example of changing the timber extraction requirement, followed by the integration of the scenario data into the hypothesis and table from Picard and Sheppard (2001b) found under Table 1.1.

3.3.1 Changing timber extraction requirements

Scenario #6 was selected as a base case to test how the model would react by changing the timber extraction requirements. Levels ranging from 40% to 95% were tested. In some cases, the
invisible cells alone were enough to satisfy the harvest requirement. In the most difficult example, a Best Percent Visible Alteration level of about 18% was met. The chart below depicts the results from this scenario as presented in Table 3.3. It is important to note that these results represent a VSU that is hypothetical and may not be the most accurate representation of a real VSU that could potentially exist.

Figure 3.28 Theoretical Timber Supply and VQO relationship with empirical example

Figure 3.29 includes plots of the best percent visible alteration solution found for each scenario. In many cases, the solution ended near 99% timber availability with a Best Percent Visible Alteration of about 1%. Therefore, many of the data are not visible at the given chart resolution. The chart clearly depicts that all scenarios, whether manually or automatically halted, found a solution provides for a significant amount of harvest. Although these may not be the most representative solutions of real-world problems, it does show that the model is creating solutions that do not have a linear relationship between the X and Y axes. Again, this graph is showing a calculation of pixel level visibility and may not best represent a full perspective level Best Percent Visible Alteration calculation. As parts of the VSU may exist in the background, and therefore
may not be included in real-world situations, the alternative method of measuring a weighted perspective percent alteration found in Appendix 9, may be more accurate.

Figure 3.29 Theoretical Timber Supply and VQO relationship with all scenario data plotted

3.4 Using Percent Visible Alteration Calculation

This section focuses on showing the results from the alternative and perhaps more precise percent visible alteration calculation. This section highlights the differences between the standard percent visible alteration calculation and the alternative method presented in Section 2.4. The primary effect difference between the two methods are that the alternative method takes a weighted approach to visibility, reducing the importance of the VSU raster and reducing the effect of visible pixels that are only visible at low angles. In the previous section (Section 3.3) the results include invisible cells which may not be an accurate real-world representation of a VSU. Yet, in the alternative method, the process removes the invisible areas from the calculation simulating a
more accurate VSU and thus more realistic outcome. The results in this section are provided in order to depict any differences between the Model’s percent visible calculation and the alternative calculation method as applied to the Scenario solutions the model derived. Only a few representative example scenarios were used to highlight the difference between the two different approaches.

3.4.1 Scenario #3

Perhaps representing the most difficult challenge, the difference between the two percent visible calculations reports some major differences. When using the integrated method within the Model versus the alternative method, the alternative method proves more accurate. The results are given in Table 3.4. The results suggest that there is not a linear relationship between the two methods. He results also show that the results produced by the integrated method are quite different than the alternative method.

<table>
<thead>
<tr>
<th>Scenario #3</th>
<th>80% harvest, 60 cell size</th>
<th>60% harvest, 60 cell size</th>
<th>80% harvest, 20 cell size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Percent Visible Alteration</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Integrated Method</td>
<td>33%</td>
<td>11%</td>
<td>14%</td>
</tr>
<tr>
<td>Percent Visible Alteration</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Alternative Method</td>
<td>50%</td>
<td>13%</td>
<td>26%</td>
</tr>
</tbody>
</table>

3.4.2 Scenario #5

This scenario was chosen as an example to see if there is any difference between the two methods when very few cells are visible. The method produced the same calculation of 1% visible alteration.

3.4.3 Scenario #6

The original VSU for this scenario includes much of the background landscape, which would most likely not be included in a real-world VSU. In addition, the scenario provides a range of timber harvest levels that allows for a better assessment of the two different methods. Only the 95% to 80% harvest levels were used to report results, since a lesser harvest rate has only a miniscule number of cells visible, even from perspective view, making the calculations similar as in the
previous example (Scenario #5). Table 3.5 is represented in graph format shown in Figure 3.30.

In this example, the alternative method results in nearly 3x of an increase in percent visible alteration.

**Table 3.5 Comparison between both PVA measurement results for Scenario #6**

<table>
<thead>
<tr>
<th>Scenario #5</th>
<th>95% harvest</th>
<th>90% harvest</th>
<th>85% harvest</th>
<th>80% harvest</th>
</tr>
</thead>
<tbody>
<tr>
<td>Percent Visible Alteration Integrated Method</td>
<td>18%</td>
<td>8%</td>
<td>3%</td>
<td>2%</td>
</tr>
<tr>
<td>Percent Visible Alteration Alternative Method</td>
<td>54%</td>
<td>24%</td>
<td>10%</td>
<td>4%</td>
</tr>
</tbody>
</table>

![Figure 3.30 Comparison between both PVA measurement results for Scenario #6](image-url)
4 Discussion

Based on the scenarios tested, the results suggest two significant findings; (1) The model has the potential for screening harvested areas within irregular landscapes, providing a method to quickly and automatically identify areas that can be harvested with little visible alteration and (2) the Model has proven to be well suited in its ability as a decision support tool to help harvest planners solve complex VRM planning problems for a variety of different landscapes. The significance of the former finding suggests that harvest planning is likely to be enhanced by the Model presented. It is acknowledge that there are limitations by using simple landscape models to test these findings, but the data showing the relationship of Best Percent Visible Alteration and timber extraction level suggested that the results are quite formidable. The latter finding suggests that the GA, in combination with the heuristics implemented, may aid in the design process on complex landscapes where both visible alteration limitations and high levels of extraction must occur in the same location.

Dealing with the more complex issues of translating a two dimensional planning problem into a three dimensional outcome can be difficult for humans because of its level of abstractness. However if a model, such as the one presented could deal with this level of complexity, it would free more time for other planning details. This key finding suggests that some level of computer automation for the VRM planning process could be used in the future. Further discussion about the two findings as well as a discussion about why certain constraints, limitations and requirements were implemented in the Model will be presented, followed by recommendations for future research.

4.1 Timber Availability, VQOs and Empirical, but Hypothetical Data

In order to balance aesthetics and timber supply, BCMoF takes into account visual resources as long as they do not excessively affect timber supply (B.C. Ministry of Forests 1996a, B.C. Ministry of Forests 1996b). Attempts to achieve this balance have been controversial primarily because the thought of managing for visual aesthetics is believed to reduce the available timber supply. In some cases this may be true, but recent research suggests quite the opposite: that partial cutting
for use in VRM can actually increase timber availability in some circumstances (Picard and Sheppard 2001b). Several other studies have looked at alternatives to clear-cutting, such as partial cutting, and show that this method increases timber availability (B.C. Ministry of Forests 1997a, Clay 1998, Fight and Randall 1980, McDonald et al. 1998). Even with the data collected from these studies more data need to be collected in order to form a better understanding of the relationship between VQO management levels and timber availability.

The data collected from the current study helps the argument that partial cutting, when used to meet certain VQO levels, can increase timber availability over clear-cutting. The data shown in Figure 3.29 provides evidence that highly restrictive VQO levels can be attained with a high percent of timber availability. However, more evidence needs to be provided in cases where extremely high slopes are present and where it is difficult to produce a screening effect. It is obvious that the scenarios, with the exception of one, may not represent realistic problems. Although there was only one realistic scenario presented, the final harvest plan produced an 80% extraction rate and fell just within the Partial Retention VQO guidance level.

In Scenario #3, when the resolution was increased from 60m to 20m, it produced the same harvest extraction level (80%) but with half the visible alteration as shown in Table 3.2. The higher resolution model also took fewer generations to complete and for a 2% drop in visible alteration, increased timber availability by 20% when compared to the 60% harvest plan in Scenario #3. This suggests that the Model has the capability to deal with increasing levels of terrain complexity and resolution. Furthermore, it suggests the potential for the Model to be run at higher resolutions, allowing for more detailed harvesting plans. Yet, at the same time if higher resolution plans are to be developed then special mechanisms that integrate operational level planning (e.g. cell adjacency) need to be considered.

Scenario #3 provides the most realistic assessment of the relationship between timber availability and visible alteration. In this Scenario, nearly the entire landscape is visible from the view point. This means that there was little opportunity for the Model to design a plan that hid harvested cells in areas invisible from the view point, but which were included in the VSU. Since the slopes are
much higher than many of the other scenarios (e.g. upwards to 55% on high slopes) it suggests that the Model can deal with the effect of planning and screening in these areas.

The Model was created to hide any part of a cut or surface deemed high contrast from the perspective view. This effect is known as the “screening effect” whereby smaller cuts or openings are selected and are hidden by a foreground of trees. These small patches have the effect of softening the visual scene, which can result in an undetectable or minimal visual impact. However, in the development stage of the Model, it was suggested that the Model be developed with some practicality in mind. The practicality condition was implemented by developing a cell adjacency structure. The combination of the adjacency structure and the “screening effect” can be detected in Figure 3.13, Figure 3.14, Figure A6.77 and Figure A6.81. The former two examples best depict the meandering effect that show pockets of connected small blocks of harvest surrounded by uncut cells which screen the blocks. The connections between these blocks have the potential to act as corridors connecting a large majority of the harvest cells.

Another outcome is the Model’s use of the foreground to hide a large portion of the background landscape. This effect is most obvious in Figure 3.1 and Figure 3.2 using a flat DEM. The cells closest to the view point (which happens to be located 3km of the top right corner of the image) show that only a few pixels along the edges are visible when cut, but that as long as the harvest is behind the front line of trees, it remains invisible. This effect is seen on almost every other scenario, so a slight increase in elevation (40m) was added granted to the view point in order to see the effects of a slightly increased view angle. Only subtle differences were noticed, primarily in scenarios where background hills or mountains were in view. The differences can be assessed by comparing the Invisible Cell results such as in Scenario #2 (Figure 3.3 vs. Figure 3.5) and Scenario #5 (Figure 3.16 vs. Figure 3.18). By increasing the view point height from 1.5m to 41.5m in Scenario #2, an increase of 12% of the landscape becomes visible, and an increase of 14% in Scenario #5 becomes noticeable after the 40m increase. These numbers may initially seem like large increases, but the 40m difference actually makes the viewpoint higher than the highest surface in Scenario #2 and almost as high as the highest surface in Scenario #5, so the effect is
not that significant. If the viewpoint was changed to a much higher elevation, then perhaps the Model would need to spend more time finding more creative ways to hide the harvest rather than merely placing it behind the tree front. However, it can be argued that vista view points, such as scenic rest stops on a road corridor, tend to be located more along the valleys of a given terrain rather than atop a mount. Despite arguments that the Model approaches design by simply hiding the harvest behind a row of trees, this process simulates what forest planner already do, but it does so automatically and with equal or more technical rigor.

The Model is able to deal with a variety of different landscapes, at a variety of different resolutions. Harvest plans are affected by terrain in order to hide cells behind hills. Harvest plans are also affected by resolution, view height and timber extraction levels. The one realistic scenario presented shows that the model can deal with more practical and complex problems than presented by the hypothetical scenarios. Thus, the tests so far, show that the capabilities of the Model can deal with real-world landscapes over a variety of different terrains and scales. The automation of this initial design phase has the potential to save planning time and could provide a way to reduce the visual impact of a harvest with very little user input.

4.2 VRM and the GA approach

The GA approach presented through the Model is similar to the many other GA-forestry related approaches in its adapted quality. None of the GA methods available in the literature (e.g. Boston and Bettinger 1999, Brookes 2001, Falcao and Borges 2001, Lu and Eriksson 2000, Stewart et al. 2004) are pure GA implementations; rather, they all have some unique modifications to fit their desired outcome. The GA approach in the Model presented in this thesis also uses a modified approach to arrive at a desired outcome. In regards to the results presented, the two most significant modifications are the adjacency requirement and initial design phase described as Step #1 and detailed in Appendix 3.1.

The adjacency requirement affected the final outcome by increasing the likelihood that all harvest cells were grouped together. This may have the effect of making the harvest design look more
like a clear cut, rather than separate clumps of cells. A subsequent drawback is that single cell
areas or tiny pockets cannot be dealt with as an autonomous unit. As shown in Figure 2.6, a
minimum of three touching cells is allowed to exist separate from the larger cut area. This
constraint was placed in the Model to provide some operational practicality. An alternative to this
approach would be to include, within the fitness value of the GA, a metric that weighted the
importance of having cells grouped together in order to reduce the cost of implementing the
harvest. However, a metric like this might have the effect of increasing the visual impact by not
allowing for the meandering effect mentioned in the previous section. The contiguous constraint
allows for a branching effect that is depicted in Figure 4.1, where harvested areas are
represented by blank cells. This effect is also depicted in Figure 3.14 and Figure A6.38, but there
were no scenarios which derived a final solution using the exact effect as shown below.

![Figure 4.1 Branching Effect](image)

As seen from the upper right corner
looking toward the bottom left
corner.

The initial design phase was implemented for two reasons, to ensure that all invisible areas were
deemed for harvest, and to increase the speed with which the model arrived at a solution. This
heuristic was created both out of necessity due to the slow nature of each generation's
processing time and to incorporate prior knowledge. For further discussion on the limitations of
implementing this Model in ArclInfo, refer to Appendix 7. Most GA implementations begin with a
population which is created by a random method. Then the GA is used to take these random
solutions and over several generations arrange the values into some resemblance of order.
However, given the technology used for this program, the luxury of beginning with a random valued population was not affordable. The processing time for each generation took between 10-20 minutes depending upon the speed of the computer used. At a minimum of 10 minute generations there would be no feasible way to run the model through 300,000 or more iterations as in the case of Falcao and Borges (2000). Furthermore, the Model was developed to modify a harvest plan, not to create a new one. Although it has the potential to modify the plan in such a way that there is little or no congruency with the initial input, the model works from an initial set of parents as depicted in Figure 2.10 and Figure 2.11.

The effect of the initial design phase on the final solution had interesting results. Generally, it is quite common for the GA to be run through thousands upon thousands of generations before arriving at or near a solution. Part of this can be explained in terms of the number of possibilities any given solution may afford. For instance, in a 25 x 25 cell grid where an 80% harvest is required, the number of potential cell arrangements is extraordinary. A permutation calculation is the best way to understand the scope and magnitude of the problem. More specifically, a combinatorial permutation is required because the spatial arrangement or order of cells does not matter. In the case of 25 x 25 grids used to test the GA, the combination calculation becomes 625 choose 500, which is a completely impractical number to try to solve by random solution generation. The GA, however, reduces the number of combinations by favoring cells that increase the best fit value. By aiding the GA and modifying the input solutions with the initial design phase the time required to find a solution is significantly reduced.

The results provide evidence that the initial design phase has reduced the number of generations required to find a solution. About ten of the solutions required fewer than 10 generations to arrive at a solution that met the user requirements. Scenarios such as Scenario #3, #4 and #6 required a greater number of iterations until completion and all were stopped manually. Looking at the chart produced from the log file output in Figure 3.15 in Scenario #4, there is a level-off effect toward the latter generations. Given the logarithmic function based on the average solutions values for each generation, there is potential for a solution with a percent visible alteration to
reach 1, but time did not allow for testing of this. However, after 245 generations the Model produced a design that would allow for an 80% harvest rate and still fall within a retention level VQO of 1.44%

4.3 Technical and Practical Issues with the Model

The final solution and result outcomes provide promising support for this model to be applicable to the harvest design process for VRM. The Model presented here offers a framework and design approach for the further development of a more practical and efficient computer program. Many bugs and problems still need to be addressed before being adapted to solve a wider array of more realistic problem. This section will address some of the current assumptions, problems, the workarounds and effects.

One of the more pertinent issues with the model is its applicability to the real world. The model provides a framework whereby operational or ecological requirements could be included by altering the fitness metric or by adding a weighted raster. As of now, the model has only been tested using the simple measurement of the number of high contrast visible cells divided by the number of cells within the VSU. One example of this would be to include economic criteria by introducing a cost surface map where the values of cells in the map might represent a predicted economic return per unit. This could then be used to multiply with the harvest plan map generated by the Model. The values would then be totaled and the Model iterated until the total reached a certain economic return balanced with visual quality or was halted.

Operationally, the model attempts to provide connectivity to cells using the cell adjacency constraint outlined in Section 2.2.4.4. The model will not automatically create roads, landings or other infrastructure. A proposed workaround is to manually modify the Target Cut and output from Step #1 by designating the spatial locations of infrastructure for harvest. The Model also assumes that there are harvest block level adjacency requirements with nearby harvested area. It is up to the user to define the patches that are available for harvest, but in accordance with partial cutting requirements on a VQO, the harvest is not subject to block adjacency constraints.
In terms of ecological effects, the model does not include any metric that evaluates or weighs the impact of the harvest for wildlife, regeneration or other related factors, but by simply applying a different metric to the fitness value the Model could be easily adapted. Landscape and ecological characteristics such as riparian zones can be figured into the model by modifying the Target Cut file to eliminate any cells that contain riparian zones. This would ensure that those cells will not be designated for harvest but will be included in the VSU.

Landscape specific elements like stand density, organic shaped designs and ridgetop considerations are dealt with indirectly. Figure 1.1 and Figure 3.24 show that a pixel was harvested at the highest point on one of the mountains. As a general landscape planning concept, harvests like these should be avoided. The user may choose to eliminate cells that are ridgetops or high slopes by modifying the Target Cut raster. In regards to organic shaped designs, the Model deals with shape because much of the design conforms to the terrain. Other characteristics like stand density or light refractivity are assumed to be consistent across the landscape.

Based on the results in Section 3.4, it would be recommended to use the alternative method of measuring the percent visible alteration. Integrating this method into the Model would be a fairly quick and easy process. It primarily entails performing the calculation before the completion algorithm is executed. This would ensure a more accurate representation of the perspective view, eliminate much of the responsibility of the user to define a VSU in 2D and enable the model to deliver more relevant results. The current calculation method works well for plans where very few cells are visible, but in more complex landscape with higher slopes, the current measurement technique would not be as effective as the alternative method.

4.4 Future Application and Development

The Model presented in this thesis has potential as a decision support tool for VRM planning. With additional tests and further development it can be adapted into a more streamlined application that can offer more robust options to meet a variety of different needs such as
ecological impacts, operational planning and economic costs associated with a harvest plan. The original scope of the Model was to include ecological and operational planning elements, but due to the high degree of ambiguity, technical expertise and time frame, the project was not completely feasible. To some degree, each of these additional elements was addressed in the Model, but more detail can be added. Using a cost surface map or changing the fitness metric to include ecological effects would greatly enhance the level of complexity the Model could deal with. This would add an entire new dimension to planning because “what-if” recommendations could be created for a variety of landscapes which harvest planners alone cannot manage. The Model developed for this thesis sets the foundation for the synthesis of the GA and GIS framework, which provide future opportunities for development.

Developing an application that works efficiently means abandoning the Model Builder interface and redeveloping the Model in a code-based interface. Most of the time involved in processing was due to the slow nature of the tools within ArcInfo. The same tools that are used in the Model are also available in a much more streamlined code-based format within ESRI’s ArcObjects. Using these tools directly in code, rather than as a model in Model Builder, may reduce the processing time for each generation down to split seconds. When development of the current Model originally started, the idea was to build a model, understand the components and test to see if further development was viable. Now that a structure is in place, the code development process can begin.

A more efficient model would allow for a broader scope of testing, both in terms of hypothetical and realistic settings. A more efficient model would allow for additional sensitivity tests to be conducted on GA variables like rate of crossover and mutation. Changing these variables might enable a quicker result or a better design. Greater levels of resolution or larger areas could also be tested, potentially allowing for both stand and landscape level problems to be designed simultaneously. Also adding elements like multiple view points and stand density would prove beneficial. To further validate the models findings, additional three dimensional renderings should be produced. Then, the Effectiveness Evaluation could be used to on these depictions to assess
if there is any difference in the percent visible alteration as measured by the Model versus measurements produced by experts.

By comparing the harvest plans from areas in BC that have been harvested under a VQO, to those plans generated by the Model presented, better and more practical assessments could be made as to the potential of the Model to act as a decision support tool. However, under the current tests, there is enough evidence to suggest that the Model could be used as a supplement to an initial design phase by quickly identifying potential plans that screen the harvested cells. Then, the plan can be manipulated by an expert who has an assortment of tools and techniques to further enhance a harvest design for visual purposes. More practical or real-world scenarios were not conducted in the current study because it was beyond the scope and time frame of this thesis and data access was limited. Additional tests could be run on regions in BC where VQOs have been established but the region has yet to be harvested. The output from these tests could be assessed in a social context using Scenic Beauty measurements and by professional planners to help understand the potential costs associated with implementing an automated recommendation. Discovering any correlations between Scenic Beauty and harvest design as well as cost factors associated with the design process will help to make the Model a more practical application capable of aiding harvest designers in creating solutions to complex real world problems.
5 Conclusion

The primary objective for this thesis was to create a model which simulates aspects of the design process and finds the “best-fit” solution for each scenario. The secondary objective, or question, was whether or not the percent of visible alteration infers a lower amount of timber availability.

By using a set of constraints, requirements, commercially available tools and custom code, a Model was developed which automates the harvest design process. The Model is capable of using a variety of input including forest cover information (tree heights), terrain data, a visual sensitivity unit and a target cut region and generating a harvest plan that attempts to meet a timber requirement and an allowable visible alteration level. Several different scenarios were used to test a variety of different land forms, including mountains, hills and valleys. The results show that within the range of landscapes tested, the Model can produce a harvest plan that meets or exceeds the VQO at a given timber harvest requirement. A few examples of 3D simulations were presented to better depict how the techniques, methods and tools used can accurately produce a harvest plan using 2D GIS data for a 3D world.

Tests were conducted with datasets in sizes ranging from 20 x 20 cell grids to 96 x 111 cell grids, using 20, 25 and 60 meter resolution. Increasing the spatial resolution appears to significantly decrease the visible alteration. Larger grids were dealt with just as quickly as the smaller grids, so the cell sizes tested had no bearing on processing time.

One of the most significant findings suggests that VRM management may actually provide a higher level of timber availability than many in the forest industry might imagine. Although the model was tested against one realistic solution, the hypothetical scenarios represented a wide variety of landscapes. According to the results, there is not a linear relationship between percent visible alteration timber availability for the first harvest period at any VQO. Additional scenarios with multiple view points should be tested to verify these findings.

The final objective was to discuss the potential for the model to be adapted as a decision support tool. As suggested in the Discussion, the Model was shown to be effective in a controlled testing
environment. Furthermore, it was shown to be applicable to various levels of landscape complexity and is easily adaptable to operational, economic and ecological elements. Additional tests should be run to confirm its viability for more realistic situations. Nevertheless, the Model does provide a solid foundation whereby a more streamlined application could be developed that would allow for a wider range of testing scenarios. The tool developed for this thesis was intended to provide a simple, practical beginning for further development of more complex and efficient model that could become adopted as decision support tool in both the research and commercial settings. The completed program can save planning time by automating the harvest design process and maximizing aesthetic quality for any given harvest level, potentially alleviating social conflicts by implementing good Visual Stewardship practices.
References


B.C. Ministry of Forests. 1998b. Procedures for factoring visual resources into timber supply analyses. Forest Practices Branch, Victoria, B.C.


Iverson, W.D. 1985. And that’s about the size of it: visual magnitude as a measurement of the physical landscape. Landscape Journal 4: 1, pg. 14-22.


Ribe, R.G. 1991. The scenic impact of key forest attributes and long term management alternatives for hardwood forests. Northeastern Forest Experiment Station.


Appendix 1

The six steps within the Genetic Algorithm are demonstrated below using simple a simple character list.

**Variables:**
- Population Size: 4
- Crossover: 0.80
- Mutation: 0.01
- Initial Population: See Time = 0, Step 1
- Fitness: The sum of each value

**Step 1**
Select initial population of solutions
- Parent A: 0 0 1 1 0 1 1 0 1
- Parent B: 1 0 0 1 1 1 0 0 1 0

**Step 2**
Determine the fitness for each solution
- Parent A = 5
- Parent B = 5

**Step 3**
Choose the best solutions based on fitness
- Parent A
- Parent B

**Step 4**
Breed the new generation through crossover of parents and mutation. In the following table, the column to the left represent the parents. The underline represents the location of the string that will be inherited in the next generation. Column two represents the crossover and the underline in this column represents the location that will be mutated. Column three is the final product of the crossover and mutation, becoming the final child.

<table>
<thead>
<tr>
<th>Parents</th>
<th>After Crossover</th>
<th>After Mutation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 0 1 1 0 1 1 0 1</td>
<td>0 0 0 1 1 0 1 1 1 0</td>
<td>0 0 0 1 1 0 1 1 1</td>
</tr>
<tr>
<td>1 0 0 1 1 1 0 0 1 0</td>
<td>1 0 0 1 1 0 1 1 1</td>
<td>1 0 1 1 0 1 1 1</td>
</tr>
<tr>
<td>0 0 0 1 1 0 1 1 0 1</td>
<td>0 0 0 1 1 1 1 1 0 1</td>
<td>0 1 0 1 1 1 1 0 1</td>
</tr>
<tr>
<td>1 0 0 1 1 1 0 0 1 0</td>
<td>1 0 0 1 1 0 1 0 0 0</td>
<td>1 0 0 1 1 0 0 1 0</td>
</tr>
<tr>
<td>0 0 0 1 1 0 1 1 0 1</td>
<td>1 0 0 1 1 0 1 1 0 0</td>
<td>1 0 0 1 1 0 0 1 0</td>
</tr>
</tbody>
</table>

**Step 5**
Determine the fitness for each solution
- Child A: 0 0 0 1 1 0 1 1 1 1 = 6
- Child B: 1 0 1 1 1 0 1 1 1 1 = 8
- Child C: 0 1 0 1 1 1 1 1 0 1 = 7
- Child D: 1 0 0 1 1 0 0 1 0 0 = 4
The example above offers a simple explanation of how the GA works, but this example is an oversimplified approach. In reality, the GA can be applied not only to string values, but to a variety of other applications. The power of the GA lies in its ability to breed the “good” qualities of each parent and be able to mutate the “poor” qualities to good qualities. Yet at the same time, the poor qualities of each parent can be reproduced and then the good qualities can be mutated to produce a child that has a significantly lower fitness than its parents. The advantage is that those children with significant gains are rewarded by being bred in the next generation. In the example above, the “randomness” applied to the crossover and mutation could have created a solution in the following generation that looks like the following:

<table>
<thead>
<tr>
<th>Parents (Child B and C)</th>
<th>After Crossover</th>
<th>After Mutation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1011101111</td>
<td>1011111111</td>
<td>1111111111</td>
</tr>
<tr>
<td>0101111101</td>
<td>0011101101</td>
<td>0011001101</td>
</tr>
</tbody>
</table>

Notice that, in this scenario, the next solution can produce a child with the highest fitness value possible and another child that loses overall fitness. However, in the case of the child that had net loss in fitness value, it was the worst possible case and still matches the fitness value of the generation previous. Although hypothetical, the example above offers a picture of how the GA works for simple string problems.
Appendix 2

The standard GA method for crossover is considered a single point crossover, but certain adaptations exist for spatially oriented, two-dimension problems. In the single point crossover method, a placeholder is located at a certain point along a genetic sequence. That same location is positioned along another genetic sequence. Then, the crossover process happens when every value to the left of the point in the first sequence is added to every value to the right of the location in the second genetic sequence. For example, take the following two genetic sequences:

Parent A: ABCDEFGHIJKLMNPQRSTUWXYZ
Parent B: ZYXWVUTSRQPONMLKJIHGFEDCBA

If the crossover is placed between the 11th and 12th value, the result for one child could be:

Child A: ABCDEFGHIKONMLKJIHGFEDCBA

Another alternative crossover method is the dual point crossover. This allows for pivots to be located at two points along a genetic sequence and then the crossover is performed. However, in a two dimension problem, the genetic sequence is structured differently. Take, for instance, a 5x5 matrix where the black cells represent the value 1 and blank cells represent the value 0:

The table is then converted into a standard string format by starting at the top left and moving to the bottom right as follows:

0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 1 1 1 0 1 1

Figure A2.1 Example Raster to string conversion

Using the single point crossover in this context merely focuses on the single dimension crossover approach. But the two dimension problem in this Model used a different approach to crossover. When analyzing the difference between a single and two dimension approaches, it is the orientation of a beginning and an end that differ. In a single dimension approach, the beginning of a string takes on the left most value and the end of a string takes on the right most value. However, in a two dimension problem, the direction from beginning to end is no longer concrete and can change dynamically. In one generation, the order could change from top to bottom, to left to right. Starting points could also start in the middle and end on the outside or vice versa. This poses a unique problem for methods that take two dimension
problems and simply convert them into single dimension problems. This approach usually forces the top left cell to be the beginning of the string.
Appendix 3

This appendix contains the processes in flow chart format as extracted directly from Model Builder. The model can be recreated with the combination of the diagrams, process documentation and code. These are provided as methodological support.
Appendix 3.1 Step #1: Find invisible areas for initial extraction
Appendix 3.2 Step #2 Assessment of Initial Population
Appendix 3.3 Step #3 Find Best Solution
Appendix 3.4 Copy Parents to Next Generation

ParentB

ParentA

Target Cut View
(zoned area for design - invisible cells)

Extract by Mask (2)

hrvstPln8

Extract by Mask

hrvstPln9
Appendix 3.5 Create Children

Reclassify so the possibility is equal to the desired mutation

Reclassify default cell size for Mutation Cell size (Calculate Value)

Create Mutation Possible (Create Random Raster)

Create Random Raster

Create Random Raster

Make crossover

Modify the raster to create the crossover (Rectify)

Take the cross over percent from parent A (Times)

Take the cross over percent from parent B (Times)

and breed them together (Single Output Map Algebra)

Mutation Starts Here

Apply the mutation (Efficient XD)

Child Honest Plan

mutation

Mutation

Mutation

rectify

crossover

rectify

crossover

rectify

crossover

rectify
Appendix 3.6 Modify Harvest Plan to Meet Extraction Requirement

- Harvest Plan Child
- Population Size
- Analysis Resolution (Cell Size)
- Temporary File Processing Folder
- Targetcut

Recalculate

prettystatu
invisiCcu

Combine the invisible areas with the visible areas for the cut design(Plus)

readyforMod

Raster to ASCII

Harvest Plan ASCII

Required Timber Extraction

Modify Input Harvest Plan (2)

Modified Harvest Plan (ASCII)

ASCII to Raster

Modified Harvest Plan (raster)

Coordinate System

Define Projection

Modified Harvest Plan (projected raster)

Calculate Value (2)

continue
Appendix 3.7 Simulate Harvest Plan

If running the model only with 1's for cut, then modify these values:

- Reclassify input data.
- Derive tree height for each cell in harvest plan.
- Simulate the harvest on the forest.
- Run viewshed to find visible areas.
- Modify the output to only give the area in the VSU.

Value to represent as false.

Modify the output so it only gives the areas in the VSU.

Calculate Value.
Appendix 4
Appendix 4.1 Step #1: Find invisible areas for initial extraction

Model Properties

Required Inputs:

- Viewpoints (viewpoints.shp)
- DEM
- Target Cut (targetCut)
- Analysis Resolution
- Coordinate System

Output:

- targetCutView
- invisibleCut

Environment Settings:

- Cell Size: Analysis Resolution
- Mask: targetCut

Process #1: Viewshed

The first order of business is to run a viewshed analysis using the viewpoints.shp and DEM. This tool generates an intermediate raster that separates the visible (value of 1) and invisible (value of 0) cells. The tool distinguishes visible from invisible cells by comparing the altitude angle of the center of each cell with the altitude angle to the local horizon (which is the viewpoint). The cell heights are calculated by using a bilinear interpolation. Detail as to how this tool works is given in 0. For details on how the viewshed works please refer to Appendix 8 and Appendix 9.

Process #2: Con

Using the Con tool available in ArcInfo, this process takes the output from Process #1 and targetCut to derive an intermediate raster that singles out the viewshed analysis only within the bounds of the targetCut. A raster called viewParentCut is the output.
Process #3: Reclassify

The previous two processes use a value of 0 for cells that are visible. However, the objective of this model is to eliminate areas that are not visible from the targetCut. So, this step takes the output from Process #2 and reclassifies it so that invisible areas (previously zero value cells) are converted to ones and all other cells are given a not data value. The output is a raster called targetModView.

Process #4: Convert to Text

This step uses the Raster to ASCII conversion tool in ArcInfo to convert the raster from Process #3 to an ASCII text file. This step is required because raster files cannot be manipulated directly in ArcInfo and must therefore be converted to an editable format. The output is an ASCII text file called targetModView.txt.

Process #5: Clean Initial Viewshed

This tool was written in Python code because no such tool was available in ArcInfo. The purpose of the tool was to find invisible cells that do not meet contiguous requirements. It is possible that some single cells may be invisible to the viewpoint, but if they are isolated from a grouping of cells (see contiguous rules for more details), they need to be removed from the invisible cells raster. Although this tampers with the idea that all non-visible cells should be harvested, it ensures a more practical approach to harvesting. Furthermore, it does not eliminate this cell from potentially being harvested as the harvest plan gets build in the Model's third step. The output is an ASCII text file called viewCutCleaned.txt. The code for this process can be found under Appendix 5.4.

Process #6: Convert back to Raster

The output from Process #5 is then converted back to raster format so that additional computations and manipulations using ArcInfo's tools can be executed. The tool that converts the
ASCII to raster is the *ASCII to Raster* tool and produces an un-projected raster image called viewCutClean.

**Process #7: Change visible areas to NoData**

Another reclassification problem, this step reclassifies the viewCutClean raster by converting all invisible areas to NoData and all visible areas to a value of 1. This raster is the foundation for the new target cut area. This becomes the bounds for all harvest design operations. The output is a raster called targetCutView. The raster still contains no projection information.

**Process #8 & #9: Define Projection**

These two processes use the same tool in ArcInfo to project viewCutClean and targetCutView to the input Coordinate System. This ensures that any processing relating to these raster files and any other raster images within the Model are consistently matched so the data's integrity is kept.

**Process #10: Change values for scripting purposes**

This is a reclassification step that takes the output from Process #6 and converts all invisible cells with a value of 1 to a value of 2. This is required for a script used on subsequent steps of the Model. The output is a raster with the targetCut extent, invisible areas as a value of 2 and all other cells as a value of 0.
Appendix 4.2 Step #2: Assessment of Initial Population

Model Properties

Required Inputs:

- Size of Initial Population
- Allowable Visible Alteration
- Required Timber Extraction
- Analysis Resolution
- Coordinate System
- Working Location
- Temporary Processing Location
- Scenario Name
- Log File Name
- VSU
- Tree Height Map (treeHeight)
- DEM
- Target Cut (targetCut)
- High Contrast Surface (hiContrastSrf)
- Viewpoints (viewpoints.shp)
- Target Cut View (targetCutView) – From Step #1, Process #8
- ParentA
- ParentB
- Invisible Cut (invisibleCut) – From Step #1, Process #10

Output:

- ParentA
- ParentB

Environment Settings:

- Scratch Workspace: Temporary Processing Location
- Cell Size: Analysis Resolution

Process #1 and #2: Extract By Mask

This step copies the two parents, ParentA and ParentB, but performs a special operation that removes any extraneous areas outside the targetCutView bounds. Notice that the targetCutView bounds and the targetCut bounds are different as the targetCutView bounds also has the invisible cells eliminated from the raster. Details of what this raster contains can be found in Step #1, Process #7. The outputs from these two steps are two harvest plans, raster images called hrvstPln0 and hrvstPln1, respectively.
Process #3: Convert Percent to Decimal

Using one of ArcInfo's new tools called Calculate Value; this step reads the Required Timber Extraction Levels and converts it to decimal. This step is required as input into Process #4, which runs a script which modifies the harvest plans. The operation for this step is as follows:

Timber Extraction Levels / 100.00

Process #4: Modify Harvest Plan to Meet Extraction Requirement

This step takes the output from Processes #1 and #2 and runs them through a model. The model alters the harvest plans so they meet both the Required Timber Extraction Levels and contiguous requirements. The output is a Boolean value representing the completion state and a raster file for each input harvest plan, called modHrvsX, where X is the number of the harvest plan input. For additional details, this step is duplicated in Step #3, Process #6.

Process #5: Simulate Harvest Plan

Using the output from Process #4, this process, which is a model, simulates the harvest on the landscape. This process uses a variety of inputs such as the DEM, Tree Height Map, High Contrast Surface, Viewpoints and others to derive new surface and visibility analysis that show areas that are now visible and have negative visual impacts. The final output is a Boolean value representing the completion state and a raster called visUglyX, where X is the number of each harvest plan. The raster contains only zeros and ones, where ones represent the surfaces that are visible and have negative visual impact. For additional details, this step is duplicated in Step #3, Process #7.

Process #6: Determine Fitness and Create Next Generation Parents

This final step is a script, written by the author, which assesses the number of cells with a value of 1 from visUglyX. The fitness is the Percent Visible Alteration (PVA), which is calculated by taking the number of cells with a value of 1 divided by the total number of cells within the raster.
Since the raster bounds are equal to the VSU, this is a fair assessment to determine if any of the solutions meet the Allowable Visible Alteration levels. This process loops through each `visUglyX`, determines the PVA, finding the two best values. These values are then correlated with the `modHrvstX` harvest plans and the two best solutions are copied as ParentA and ParentB as raster output. The other output is a variable called Best Percent Visible Alteration, which is the PVA of ParentA. The code for this process can be found under Appendix 5.2.
Appendix 4.3 Step #3: Find the Best Solution

Model Properties

Required Inputs:

Population Size
Start Generation Number
Maximum Number of Iterations
Size of Initial Population
Allowable Visible Alteration
Required Timber Extraction
Analysis Resolution
Coordinate System
Working Location
Temporary Processing Location
Scenario Name
Log File Name
VSU
Tree Height Map (treeHeight)
DEM
Target Cut (targetCut)
High Contrast Surface (hiContrastSrf)
Viewpoints (viewpoints.shp)
Target Cut View (targetCutView) – From Step #1, Process #8
ParentA – From Step #2, Process #6
ParentB – From Step #2, Process #6
Invisible Cut (invisibleCut) – From Step #1, Process #10

Output:

ParentA
ParentB

Environment Settings:

Scratch Workspace: Temporary Processing Location
Cell Size: Analysis Resolution

Process #1: Calculate Value

One of the major deviations from the typical GA approach is that the two parents from the previous generation are copied to form part of the next generation’s children. This could allow for a pseudo-immortal solution as long as that solution has a higher fitness than any of its children. This will be further discussed in Process #2 and #3, but for now, the reason for Process #1 is to determine the actual number of children that need to be generated. Taking the user input
variable, Population Size, and subtracting by two, the number of parents, the Number of Children is derived as output.

**Process #2 and #3: Copy Parents to Next Generation**

This process is a model that contains two processes and requires ParentA, ParentB, Analysis Resolution, Target Cut View and input. The outputs are two raster files which are called modHrvst9 and modHrvst8. These names are manually set as parameters by the user and are dependent upon the Population Size. So, if the population size is ten, these names would be used. Alternatively, the names for these files should be as such: modHrvst(n-2) and modHrvst(n-1) where n is the Population Size. A diagram of this sub process can be found under Appendix 1.1.

The purpose for this process is to ensure that these two solutions, which are the best solutions from last generation, are not tampered with. One reality of using a model that takes a long time to process is that any advancement toward a solution should not be compromised. By securing these parents as children in the current solution and not modifying them, the model ensures that, at worst, no solution will have a lower Percent Visible Alteration than the previous generation.

**Process #2.A: Extract By Mask**

This step simply takes ParentA and removes from the solution the areas that are invisible by using Target Cut View as the mask. The output is a raster image with the bounds of Target Cut View. This becomes the tenth child solution in the generation.

**Process #2.B: Extract By Mask**

This step simply takes ParentB and removes from the solution the areas that are invisible by using Target Cut View as the mask. The output is a raster image with the bounds of Target Cut View. This becomes the ninth child solution in the generation.
Process #4: Create Children

Yet another model, this process contains 12 steps and produces raster files that are bred from ParentA and ParentB using a series of tools and mechanisms. This model requires the Analysis Resolution, Target Cut View, ParentA, ParentB, and the Temporary File Processing input. The outputs are raster images called hrstPlnX, where X is the Number of Children. The environment settings for this raster are a Scratch Workspace of the Temporary File Processing and a Mask of the Target Cut View raster. The model loops for each child and creates a different set of raster images throughout Process #4.B - #4.K. A diagram of this sub process can be found under Appendix 3.5.

Process #4.A: Get the Working Cell Size

This process uses the Get Raster Properties tool from ArclInfo, which gets a variety of properties from any raster image. This process gets the cell size from ParentA and uses it as the default cell size for all the analysis within Process #4. The output is Cell Size.

Process #4.B: Recalculate Default Cell Size for Mutation Cell Size

Using the Calculate Value tool, this process multiplies the Cell Size from Process #4.A times three. This becomes the new cell size for the mutation raster to be generated. The reason for changing the cell size for a mutation raster is because the contiguous requirements state that no single cell can exist as a standalone cell. So, in order to encourage the mutation while meeting contiguous requirements, larger mutation blocks (cell size times 3) need to be created to ensure a better likelihood of a mutation not being eliminated by contiguous requirements.

Process #4.C: Create Mutation Possibility

This process uses the tool Create Random Raster to generate a random raster with the cell size from Process #4.B. The raster cell values range between 0 and 1 and use a
uniform distribution so that each cell has equal opportunity of being assigned a specific value between 0 and 1.

Process #4.D: Reclassify so Possibility is Equal to Desired Mutation

Using the reclassification tool, this process reclassifies the raster from Process #4.D. This step simulates a GA mutation, but does so in a spatial format. Cell values from 0-.949999 become 0 and cell values from .949999-1 become 1. This calculates a mutation rate of 5%, but because the original raster was created using random value, a mutation is not always guaranteed because all values could potentially fall between 0 and .949999. The final output is a raster, called mutateRaster, which hopefully contains 5% of the cells with a value of 1 and 95% of cells with a value of 0. This step also deviates from the raw form of the GA. It is an initial attempt to meet both contiguous requirements, while encouraging mutation and doing so in such a manner that the mutation is spatially set.

Process #4.E: Create Random Raster

This process creates a random raster, called crossoverUni, which is similar to Process #4.C but with the cell size from Process #4.A. This raster will be used as the crossover map.

Process #4.F: Modify the Raster to Create the Crossover

Using the reclassification tool, this process reclassifies the cell values from the raster generated in Process #4.E. The tool maps any cell value from 0-.799999 as 0 and .799999-1 as 1. The final outcome is a raster, called crossover, consisting of ones and zeros randomly placed throughout the grid. In the genetic algorithm, this step set the crossover rate.
Process #4.G: Reclassify

This tool takes the raster generated from Process #4.F and inverses the values to produce a new raster called crossoverInv. The combination of this raster and the raster from Process #4.F form the spatial crossover.

Process #4.H: Take the Crossover Percent from ParentA

This step multiplies the raster from Process #4.F with ParentA to form a new raster called partA, which contains a copy of roughly 80% of ParentA.

Process #4.I: Take the Crossover Percent from ParentB

This step multiplies the raster from Process #4.F with ParentB to form a new raster, called partB, which contains a copy of roughly 20% of ParentB.

Process #4.J: And Breed them Together

Using the Single Output Map Algebra tool available in ArcInfo add partA and partB together using the formula max(partA, partB) to form a "clean" child called newhrvstPln. This child is without mutation.

Process #4.K: Apply the Mutation

Using a special map algebra tool Bitwise XOR, take the output from Process #4.J and #4.D and apply the mutation. The Bitwise XOR converts any cell value equal to 1 from newhrvstPln that match with the same cell value in mutateRaster to 0. It also converts any cell value equal to 0 in newhrvstPln where the cell value in mutateRaster is 1, to 1. This process creates the output solution called hrvstPlnX where X is the number of children.
Process #4.L: Calculate Value

The final process just assigns the current child to a variable called continue. This ensures that the next process can continue.

Process #5: Convert Percent to Decimal

The input, Required Timber Extraction Levels, is entered by the user in double format, with a range from 0 to 100, which represents a percentage value. However, to be used by the Model, this value needs to be converted to a decimal value between 0 and 1. The Calculate Value tool available in ArcInfo allows for this value to be created quickly. By taking the Required Timber Extraction Level and dividing it by 100.00, a decimal value is output.

Process #6: Modify Harvest Plan to Meet Extraction Requirement

Perhaps the most intelligent component of the entire Model, this process is built around a script that modifies the harvest plans from Process #4.K to meet both contiguous and timber extraction requirements. This process is yet another model that takes the harvest plans from Process #4.K, the Population Size, Analysis Resolution, Temporary File Processing Folder, Target Cut, invisibleCut, decimal format of Required Timber Extraction Levels from Process #5 and a Coordinate System as inputs. The environment extent is set as the same as Target Cut. The final output is a raster file for each harvest plan, called modhrvstX, where X is the population size. Each process within this model is looped for each input harvest plan so that the final modification plans are all unique. A diagram of this sub process can be found under 0.

Process #6.A: Reclassify

The first order of business is to take the harvest plans derived from Process #4.K and ensure that there are no NoData values, by assigning them to a value of 0. Doing this effectively assigns the area of the harvest plan that is equivalent to the invisibleCut to a value of 0. Since these harvest plans were given the bounds of targetCutView, which is only those areas that are visible or invisible and did not meet the contiguous
requirements, this step converts those areas to a value of 0. This allows for future mathematical operations without jeopardizing those areas with NoData.

**Process #6.B: Combine the Invisible Areas with the Visible Areas for the Cut Design**

Using the output from Process #6.A and combining the raster invisibleCut using the Plus tool, an output raster called, readytoMod is created. This raster contains three values; twos for cells that are invisible, ones for cells that are designated as part of the harvest plan and 0, all other cells.

**Process #6.C: Raster to ASCII**

Once again, before manipulating the raster, it needs to be converted to an ASCII format because ArcInfo does not allow direct manipulation of individual cell values. This tool converts the raster from Process #6.B and prepares it for manipulation in the next process.

**Process #6.D: Modify Input Harvest Plan**

This process could be called the brains of the operation. Where all the rest of the tools and processes either simulate or create the harvest plan structure, this process actually ensures that all requirements and constraints are met. This process is also a step away from the GA approach because it modifies the solution without allowing it to be run through a fitness value selection first. However, because these harvest plans need to meet certain requirements and there are a limited number of generations that can be run, this step greatly reduces the amount of processing time by eliminating iterations.

The process is a script that was generated by the author of this thesis. All of the code was manually written and is original. The code uses a complex set of validation rules to ensure that contiguous requirements are met. It also uses a simple algorithm to ensure
that the timber harvest extraction levels are met. The foundation of this code is as follows:

1. Find and fix all cells so the harvest plan meets contiguous requirements

This operation loops through the ASCII file starting at the top left, ending with the bottom right value, and queries each cell to verify that it meets the contiguous requirements. It does this by looking at the value for each neighboring cells. If there exists at least two neighboring cells with a value of 1 or 2, in such an arrangement as required by the contiguous rules, then the cell is left alone. Alternatively, if the cell does not have at least two neighboring cells with a value of 2 or 1 or the cell do not meet the contiguous requirement, then the cell is changed to a value of 0. The relevant code can be found in Appendix 5.3 under the method cleanChild().

2. Assess the harvest plan for the current harvest levels, if the harvest plan meets the required amount, then exit the script

This step simply counts the number of cell values with either a 1 or 2, and compares it to total number of cells within the targetCut bounds times the required timber extraction levels; if the values are equal, then the program stops.

3. Determine if fewer or additional cells need to be added to the harvest plan

If the values are not equal, determine the direction of the difference. If more cells need to be added, then an “increase” value is passed to a method called modifyValue(), alternatively a value of “decrease” is passed.

4. Create a map of all cells determining if they can be modified and, if so, added as part of the harvest plan or removed from the current harvest plan

To minimize the processing time, part of this step is actually integrated with the second step. What happens is that as the second step loops through the file from top left to
bottom right, it determines if a cell is equal to 0. If the cell is equal to 0, then it also verifies that if the value is changed to 1 (added to the harvest design), that it would meet the contiguous requirements. This is done by calling the method validPixelToChange (Current Row, Current Column, and direction of modification [e.g. increase or decrease]). When a cell is determined to meet these criteria, it is added to a map of cells that a deemed changeable in the increasing direction.

The other half of this step happens when there are too many cells allocated in the harvest plan. In this situation, a map of all cells with a value of 1 is created, regardless if eliminating the cell would harm the integrity of the contiguous rules. Reducing the number of cells designated within a harvest plan, while maintaining contiguous requirements, was found to be the most difficult step of all. During the development phase of this script, certain tests revealed that three cells could exist that met the minimum contiguous requirement and therefore could not be modified; but at the same time, at least one of these cells needed to be removed in order to meet the timber harvest requirement. To get around this problem, without spending a great amount of time dealing with recursion methods, memory problems and artificial intelligence, each pixel with a value of 1 had the same opportunity of being designated as a potential cell to change. The effect was an algorithm that penalized the small remote harvest blocks because removing a cell from that area has the same likelihood of removal from the larger harvest blocks, yet the small blocks have a harder time meeting the contiguous requirements because fewer neighboring cells are available.

5. Reduce or increase the number of cells designated for harvest depending upon the extraction requirements

This step uses either of the two maps, depending upon if more or fewer cells are required within the harvest plan. If fewer cells are required, then for each cell that needs to be eliminated, a random cell from the map is selected, removed from the map and converted to a value of 0. If additional cells are needed, then a random cell is selected from the map.
created in the second step, removed from the map and changed to a value of 1. Next, the map is then updated by verifying if the new addition to the harvest plan created a situation where any neighboring cells with a value of 0 is now changeable. This is done in the same format as explained in the first paragraph of the previous step.

6. Loop to the first step

The final step is a loop to the first step to guarantee any modifications to the plan to meet contiguous requirements.

When the number of cells required for harvest meets the number of cells to be harvested (either a value of 1 or 2), then the script ends and outputs a text file called modhrvstX.txt where X is the child number 0 through Population Size. The code for this process can be found under Appendix 5.3.

Process #6.E: ASCII to Raster

The output from Process #6.D is then converted back to raster format so additional computations and manipulations using ArcInfo's tools can be executed. The tool that converts the ASCII to raster is the ASCII to Raster tool and produces an un-projected raster image called modHrvstX, where X is the child number from 0 to Population Size.

Process #6.F: Define Projection

This process uses ArcInfo's Project tool and ensures that the new raster from Process #6.E is converted to the input Coordinate System. This ensures that any processing relating to these raster files and any other raster images within the Model are consistently matched so that the data's integrity is kept.
Process #6.G: Calculate Value

A calculation that returns the current iteration (child being process) is returned. This enables for the continuation of the Model once modifications for each child is complete.

Process #7: Simulate Harvest Plan

This process, yet another model, takes the output from Process #6 and simulates the harvest plan actually happening on the landscape. Inputs into this model are: Population Size, analysis Resolution, Temporary Processing Folder, the output from Process #6 (modHrvstX), the Tree Height Map, DEM, High Contrast Surface, Viewpoints, and the VSU. The environment variables set the extent as the same as the DEM, and the cell size as the Analysis Resolution. The model consists only of ArcInfo’s tools and keeps strictly to raster images. The final output is a raster image that determines the cells which are visible and considered to have a negative visual impact. A diagram for this sub process can be found under Appendix 3.7.

Process #7.A: Reclassify

Using the output from the previous Process, this tool converts all cells with a value of 2 (invisible cells deemed for harvest) and converts them to a value of 1. It also takes all NoData cells and converts them to a value of 0. This reclassification is for preparation of ensuring that the viewshed analysis will work properly because the fewer NoData cells the better the results. It also converts the modHrvstX raster so that it matches the extent of the DEM.

Process #7.B: Combine All the Ground Surfaces

This step takes the harvest plan from Process #7.A and the High Contrast Surface map and merges them together to form a raster that contains all ground level surfaces that, if seen, would have negative visual impacts. To get this output, a tool called Bitwise Or is ran to find all cells from either input raster that are equal to a value of 1. The output is a
raster called uglySurface, which suggests that the cells with a value of 1 are not visually appealing.

**Process #7.C: Derive treeHeight for Each Cell in harvestPlan**

Multiplying the treeHeight raster and the raster from Process #7.A, hrvstX returns a new raster with cell values of 0 and cell values of treeHeight, where a cell value of 1 in hrvstX exists. The output raster is called harvestPlanHt. This raster gets all the heights of the trees, or cells, within the harvest plan.

**Process #7.D: Create the Forest By Adding treeHeight to DEM**

This process takes the Tree Height Map and adds it to the DEM using the tool *Plus*. This creates a new map that simulates the actual vegetation cover on top of the ground. The final output is a raster called forestland.

**Process #7.E: Simulate the Harvest on the Forest**

This process takes the output from Process #7.D and simulates the cut by subtracting the tree heights from the harvest plan area using the output from Process #7.C. The output is a new raster that reflects the simulate harvest on the landscape.

**Process #7.F: Run Viewshed to Find Visible Areas After Cut**

The next order of business is to run a viewshed analysis using the viewpoints.shp and the simulate cut raster from Process #7.E. This tool generates a raster, called viewedCut, that separates the visible (value of 1) and invisible (value of 0) cells. The tool determines visible from invisible cells by comparing the altitude angle of the center of each cell with the altitude angle to the local horizon. The cell heights are calculated by using a bilinear interpolation. Detail as to how this tool works is given in Appendix 8. The combination of Process #7.E and this process captures the power of modeling. Being able to simulate a harvest and "see" the effects without causing any change to the actual
forest cover enables testing a variety of harvest plans on the same landscape and assesses each one in order to find the best solution. Although this is not a perfect reproduction of the "real world", it does offer a few key hints as to the visual effect of the harvest plan.

**Process #7.G: Con**

This process uses the Con tool to determine the intersection of all cell values of 1 from Process #7.B and Process #7.F. This produces an output raster that finds all surfaces that are visible and have properties that negatively affect the view as seen from the viewpoint. The output is a raster called visUgly.

**Process #7.H: Modify the Output so it Only Gives the Area in the VSU**

This process multiplies the VSU raster and the visUgly raster. Doing this converts all cells in the visUgly raster to NoData, where they do not exist within the VSU. The intention is to create a raster that returns only cells within the VSU so the final Percent Alteration Value can be calculated by dividing the number of visible cells with negative visual impacts by the total number of cells in the VSU. The output is a raster for each child, called visUglyX, where X is the child's number 0 through Population Size.

**Process #7.I: Calculate Value**

A calculation that returns the current iteration (child being process) is returned. This enables for the continuation of the Model once modifications for each child is complete.

**Process #8: Get the Current Iteration Number**

Using the Calculate Value tool returns an integer value of the Start Generation Number plus the current iteration number.
**Process #9: Copy Generation Solution Files**

This process, a script written by the author of this thesis, takes input from Process #8 and the Working Location, saves a copy of all the output solution files and visUglyX solutions and copies them to a folder within the Working Location called GenerationX, where X is the integer number from Process #8. This is a key feature that allows for exploration into how the solutions changed over time. It is also a backup that allows for a process to be restarted at any generation that has already been processed. The code for this process can be found under Appendix 5.1.

**Process #10: Determine Fitness and Create Next Generation Parents**

This process, another script written by the author of this thesis, takes as input the Log File Name, the Working Location, Required Extraction Levels, Scenario Name, Allowable Visible Alteration (AVA) and the Population Size. It also requires that modHrvstX and visUglyX are present within the Working Location. This step is fairly straightforward and relies solely on tools available within ArcInfo to read raster table information, delete old Parent solutions and copy new solutions. The script gets the number of cell values equal to 1, for each visUglyX, and divides it by the total number of cells to arrive at a Percent of Visible Alteration (PVA). The script loops through each visUglyX raster and finds the lowest PVA, then assigns the related modHrvstX raster images as ParentA and ParentB. If the user selected, the script also records the PVA values for each visUglyX within a log file. The final outputs are two raster files, ParentA and ParentB, and a value with the best PVA from ParentA. The code for this process can be found under Appendix 5.2.

**Process #11: Determine If a Solution Has Been Found**

The final process in step #3 is to determine if a solution has been found or exit the step if the Maximum Number of Iterations has been exceeded. Using the tool Calculate Value the process compares the AVA and the PVA. If the PVA is equal to or less than the AVA, then the model is stopped, otherwise it tells the model to loop back to Process #1.
Appendix 5

Appendix 5.1 Copy Database

# This script copies the current generation file for a historical record of all generations
#
#
#Import Modules
import sys, string, os, shutil, arcgisscripting

# Create the Geoprocessor object
gp = arcgisscripting.create()

#Import Variables

iterationNum = sys.argv[1]
Working_Location = sys.argv[2]

gp.AddMessage("Copying the solution to " + Working_Location[:-1] + "\Generation" + iterationNum)

try:
    #Source and destination locations
    srcName = Working_Location[:-10]
    dstName = Working_Location[:-15] + "\Generation" + iterationNum
    shutil.copytree(srcName, dstName)
except:
    gp.AddMessage("Failed to copy")

gp.SetParameterAsText(2, "True")
Appendix 5.2 Determine Fitness and Create New Parents

# Import system modules
import sys, string, os, arcgisscripting, math, time

# Create the Geoprocessor object
gp = arcgisscripting.create()

# Load required toolboxes...
gp.AddToolbox("C:/Program Files/ArcGIS/ArcToolbox/Toolboxes/Data Management Tools.tbx")
gp.AddToolbox("C:/Program Files/ArcGIS/ArcToolbox/Toolboxes/Data Management Tools.tbx")

# Calculates the percent visible alteration (PVA)
def getPVA(inRaster):
    typeOfValue = 0
    try:
        rasTable = gp.SearchCursor(inRaster) # place the table in memory
        rasRow = rasTable.Next() # reads first row where value = 0
        totalNumCells = rasRow.GetValue('Count') # gets the total number of cells for value = 0
        typeOfValue = rasRow.GetValue('Value') # Gets weather the value was one or zero
        try:
            rasRow = rasTable.Next() # reads second row where value = 1
            totalNumVisible = rasRow.GetValue('Count') # gets the value from count in the second row of
            the table
            totalNumCells = totalNumCells + totalNumVisible # now we have the total number of cells in
            the raster
        except:
            # The raster only contains one value, zero or one so only the first row will return values
            if typeOfValue == 1:
                totalNumVisible = totalNumCells # Since all pixels were visible, that need to be reported
            else:
                totalNumVisible = 0

        # This operation will return four significant digits (e.g. 5.24% or .0524)
        PVA_sig = math.ceil(float(totalNumVisible) / float(totalNumCells)) * 100000000
        PVA_result = PVA_sig/1000000 # This is percent not decimal representation
        # return the PVA
        return PVA_result
    except:
        gp.AddMessage("Failed getPVA")

# Parameter Variables...

###

133
Delete_Succeeded = "false"

Population_Size = sys.argv[1] #the number of children in the population, this will help make the
visUgly1, 2, 3 ... n

#Working Directory or GeoDatabase
Working_Location = sys.argv[2]

Viewable_Ugly_Surface = Working_Location[:-9] + "visUgly"
Harvest_Plan = Working_Location[:-9] + "modhrvst"

#The output variables
ParentA = sys.argv[3]
ParentB = sys.argv[4]
Best_Percent_Visible_Alteration = 100

Required_Timber_Extraction_Level = sys.argv[5]
Allowable_Visible_Alteration = sys.argv[6]
Scenario_Name = sys.argv[7]

#writeToLogFile = true
Log_File = sys.argv[8] #A boolean True/False checkbox, used to record PVA results
to gp.AddMessage(Log_File)
Log_File_Name = sys.argv[9] #A directory location where the log file is stored

try:

    #Loop through each child's visible surface and chose the top two solutions
    for i in range(int(Population_Size)):
        gp.AddMessage(Harvest_Plan + str(i))
        PVA = getPVA(Viewable_Ugly_Surface + str(i))

        #Check to see if the calculated PVA is less than the current Best PVA
        if float(PVA) < float(Best_Percent_Visible_Alteration):
            Best_Percent_Visible_Alteration = PVA

    try:

        #gp.AddMessage(Log_File)
        #Record the result to a log file if the user chose to log the results
        if Log_File == "true":
            gp.AddMessage("Recording Log File")
            #This is a log file that records each value from every possible solution
            recordFile = open(Log_File_Name, "a")
            if int(i) == 0:
                recordFile.write("Scenario_Name: " + str(Scenario_Name) + " > Extract: " + Required_Timber_Extraction_Level + ", AVA: " + Allowable_Visible_Alteration + ", Generation Completed At: " +
                time.strftime("%Y/%m/%d %H:%M") + ")
            recordFile.write("PVA: " + str(PVA))
            recordFile.close()
except:
gp.AddMessage("Could not write to log file")

if int(i) == 0:  # The first child should be set to the first Parent
    tempPVA_A = PVA
    tempRaster_A = Harvest_Plan + str(i)
elif int(i) == 1:  # The second child should be set as either the first or second Parent
    if PVA < tempPVA_A:
        # Set Parent A to Parent B
        tempPVA_B = tempPVA_A
        tempRaster_B = tempRaster_A
        # Set this child as Parent A
        tempPVA_A = PVA
        tempRaster_A = Harvest_Plan + str(i)
    else:
        # Set this child to Parent B
        tempPVA_B = PVA
        tempRaster_B = Harvest_Plan + str(i)
else:
    # Now ensure that we have the best solutions as Parents
    if PVA < tempPVA_A:
        # Set Parent A to Parent B
        tempPVA_B = tempPVA_A
        tempRaster_B = tempRaster_A
        # Set this child as Parent A
        tempPVA_A = PVA
        tempRaster_A = Harvest_Plan + str(i)
    elif (tempPVA_A < PVA < tempPVA_B):
        # Set this child to Parent B
        tempPVA_B = PVA
        tempRaster_B = Harvest_Plan + str(i)
gp.AddMessage("Success! Next generation's parents have been found")
newParents = "True"
except:
gp.AddMessage("The script that determines the two parents failed.")
newParents = "False"

if newParents == "True":
    try:
        gp.Delete_management(ParentA, "RasterDataset")
        gp.Delete_management(ParentB, "RasterDataset")
        Delete_Succeded = "True"
    except:
        Delete_Succeded = "True"

    gp.AddMessage("The lowest Percent Visible Alteration is: " +
    str(Best_Percent_Visible_Alteration))
    ParentA = gp.CopyRaster_management(tempRaster_A, ParentA, "", "", "", "NONE", "NONE",
                                          "")
    ParentB = gp.CopyRaster_management(tempRaster_B, ParentB, "", "", "", "NONE", "NONE",
                                          "")
    except:
        gp.AddMessage("Failed to delete the previous Parents")
#This returns the output value for the best PVA
#In the parameters window, this is the derived value output
#Note: that the parameter must be the item # as shown below

gp.SetParameterAsText(9, Best_Percent_Visible_Alteration)
Appendix 5.3 Modify Harvest Plan to Fit Contiguous Requirements

# Import system modules
import sys, string, os, arcgisscripting, math, time

# Create the Geoprocessor object
gp = arcgisscripting.create()

# Load required toolboxes...
gp.AddToolbox("C:/Program Files/ArcGIS/ArcToolbox/Toolboxes/Data Management Tools.tbx")
gp.AddToolbox("C:/Program Files/ArcGIS/ArcToolbox/Toolboxes/Data Management Tools.tbx")

# Calculates the percent visible alteration (PVA)
def getPVA(inRaster):
    typeOfValue = 0
    try:
        rasTable = gp.SearchCursor(inRaster) # place the table in memory
        rasRow = rasTable.Next() # reads first row where value = 0
        totalNumCells = rasRow.GetValue('Count') # gets the total number of cells for value = 0
        typeOfValue = rasRow.GetValue('Value') # gets weather the value was one or zero
        try:
            rasRow = rasTable.Next() # reads second row where value = 1
            totalNumVisible = rasRow.GetValue('Count') # gets the value from count in the second row of the table
            totalNumCells = totalNumCells + totalNumVisible # now we have the total number of cells in the raster
        except:
            # The raster only contains one value, zero or one so only the first row will return values
            if typeOfValue == 1:
                totalNumVisible = totalNumCells # Since all pixels were visible, that need to be reported
            else:
                totalNumVisible = 0

    except:
        gp.AddMessage("Failed getPVA")

    # This operation will return four significant digits (e.g. 5.24% or .0524),
    PVA_sig = math.ceil((float(totalNumVisible) / float(totalNumCells)) * 100000000)
    PVA_result = PVA_sig/1000000 # This is percent not decimal representation

    # return the PVA
    return PVA_result

# Parameter Variables...
###
#######
# IF CHANGING THE INPUTS/OUTPUTS, make sure to change the SetParameterAsText at the bottom of the script

Delete_Succeeded = "false"

Population_Size = sys.argv[1]  # the number of children in the population, this will help make the visUgly1, 2, 3 ... n

# Working Directory or GeoDatabase
Working_Location = sys.argv[2]

Viewable_Ugly_Surface = Working_Location[:-9] + "visUgly"
Harvest_Plan = Working_Location[:-9] + "modhrvst"

# The output variables
ParentA = sys.argv[3]
ParentB = sys.argv[4]
Best_Percent_Visible_Alteration = 100

Required_Timber_Extraction_Level = sys.argv[5]
Allowable_Visible_Alteration = sys.argv[6]
Scenario_Name = sys.argv[7]

# writeToFile = true
Log_File = sys.argv[8]  # A boolean True/False checkbox, used to record PVA results
to gp.AddMessage(Log_File)
Log_File_Name = sys.argv[9]  # A directory location where the log file is stored

try:
    # Loop through each child's visible surface and chose the top two solutions
    for i in range(int(Population_Size)):
        gp.AddMessage(Harvest_Plan + str(i))
PVA = getPVA(Viewable_Ugly_Surface + str(i))

    # Check to see if the calculated PVA is less than the current Best PVA
    if float(PVA) < float(Best_Percent_Visible_Alteration):
        Best_Percent_Visible_Alteration = PVA

    try:
        # gp.AddMessage(Log_File)
        # Record the result to a log file if the user chose to log the results
        if Log_File == "true":
            gp.AddMessage("Recording Log File")
            # This is a log file that records each value from every possible solution
            recordFile = open(Log_File_Name, "a")
            if int(i) == 0:
                recordFile.write("I")
            recordFile.write("\n")
        recordFile.write(Scenario_Name + " > Extract: " + Required_Timber_Extraction_Level + ", AVA: " + Allowable_Visible_Alteration + ", Generation Completed At: " + time.strftime("%y/%m/%d %H:%M") + ")
        recordFile.write("I")
        recordFile.close()
except:
gp.AddMessage("Could not write to log file")

if int(i) == 0:  #The first child should be set to the first Parent
    tempPVA_A = PVA
    tempRaster_A = Harvest_Plan + str(i)
elif int(i) == 1:  #The second child should be set as either the first or second Parent
    if PVA < tempPVA_A:
        #Set Parent A to Parent B
        tempPVA_B = tempPVA_A
        tempRaster_B = tempRaster_A
        #Set this child as Parent A
        tempPVA_A = PVA
        tempRaster_A = Harvest_Plan + str(i)
    else:
        #Set this child to Parent B
        tempPVA_B = PVA
        tempRaster_B = Harvest_Plan + str(i)
else:  #Now ensure that we have the best solutions as Parents
    if PVA < tempPVA_A:
        #Set Parent A to Parent B
        tempPVA_B = tempPVA_A
        tempRaster_B = tempRaster_A
        #Set this child as Parent A
        tempPVA_A = PVA
        tempRaster_A = Harvest_Plan + str(i)
    elif (tempPVA_A < PVA < tempPVA_B):
        #Set this child to Parent B
        tempPVA_B = PVA
        tempRaster_B = Harvest_Plan + str(i)
    else:
        gp.AddMessage("Success! Next generation’s parents have been found")
        newParents = "True"
except:
gp.AddMessage("The script that determines the two parents failed.")
newParents = "False"

if newParents == "True":
    try:
        try:
            #Try deleting the files, if they aren’t there then exit the try except gracefully
            gp.Delete_management(ParentA, "RasterDataset")
            gp.Delete_management(ParentB, "RasterDataset")
            Delete_Suceded = "True"
        except:
            Delete_Suceded = "True"
    except:
        gp.AddMessage("Failed to delete the previous Parents")
        Delete_Suceded = "False"

    gp.AddMessage("The lowest Percent Visible Alteration is: " + str(Best_Percent_Visible_Alteration))
    gp.AddMessage("Creating parents...")
    ParentA = gp.CopyRaster_management(tempRaster_A, ParentA, "", "", "", "NONE", "NONE", ""
    ParentB = gp.CopyRaster_management(tempRaster_B, ParentB, "", "", "", "NONE", "NONE", ""

    except:
        gp.AddMessage("Failed to delete the previous Parents")
# This returns the output value for the best PVA
# In the parameters window, this is the derived value output
# Note: that the parameter must be the item # as shown below

gp.SetParameterAsText(9, Best_Percent_Visible_Alteration)
Appendix 5.4 Modify Initial Viewshed to Fit Contiguous Requirements

# modify2FitContiguousRequirement.py
# Created by Brent Chamberlain

# Import system modules
import sys, string, os, arcgisscripting, math, random
from array import array

# Create the Geoprocessor object
GP = arcgisscripting.create()

GP.AddMessage("Attempting to modify the design to meet the timber extractions requirements...")

# This method reads in values to an array and sums the total to get a count
def readlnValues(inText):
    # In order for the validPixel to work I need to add a one cell buffer around the entire raster to ensure that it works.
    # Perhaps one way to get around this is to add a -9999 value to the bottom, top, left and right of the raster.
    # In order to make this work, I would need to alter how the random number generator is configured

    # Open original file, copy it to the new file and keep the new file open for writing
    inFile = open(inText, V)
    numRow = 0

    # Instantiate the row counter
    numRow = 0

    # Read through the ASCII file and copy the contents to variables
    while 1:
        line = inFile.readline()
        if not line: break

        # Copy the header information into a different array for easy search and replace
        if numRow <= 5:
            headerInfo.append(line)
            numRow = numRow + 1
        else:
            # This is where the buffer around the entire raster is built to allow the validPixel verifier to work without too much error checking
            if numRow == 6:
                # Then create a buffer on each side of the current line's values
                line = "-9999 " + line
                line = line + " -9999"
            pixelValues.append(line.split())

            for i in range(len(pixelValues[0])):
if i == 0:
    templine = "-9999 
else:
    templine = templine + "-9999 
pixelValues.insert(0, templine.split()) #this adds the buffer line

numRow = numRow + 1 #add another row so that it doesn't = 6 and stay in this condition
else:
    line = "-9999 " + line.
    line = line + "-9999"
    pixelValues.append(line.split())

#place in the last buffer line
for i in range(len(pixelValues[0])):
    if i == 0:
        templine = "-9999 
    else:
        templine = templine + "-9999 
    pixelValues.append(templine.split())

#close the inFile because all the contents are stored in an array
inFile.close()

#This method will remove any pixels that do not meet the continuity requirements.
#This ensures a clean solution each time.
def cleanChild():
    for i in range(1, len(pixelValues) - 1):
        #don't read the first or last line because it is a buffer
        for j in range(1, len(pixelValues[i]) - 1):
            #don't read the first or last column because it is a buffer
            if pixelValues[i][j] == "1":
                if validPixelToChange(i, j, "increase") == 0:
                    #If the pixel doesn't meet adjacency requirements
                    pixelValues[i][j] = "-9999" #Set the pixel value to NoData
                    GP.AddMessage("Changed Puxel Value")

    #Get the total number of cells planned for harvest
def getCount():
    
    #the variable list for the method
    i = 0 #counter first dimension
    j = 0 #counter second dimension
    totalCount = 0 #contains the total sum of all cells
    totalAvailableCells = 0

    for i in range(1, len(pixelValues) - 1):
        #don't read the first or last line because it is a buffer
        for j in range(1, len(pixelValues[i]) - 1):
            #don't read the first or last column because it is a buffer
            if pixelValues[i][j] == "1":
                totalCount = totalCount + 1
                totalAvailableCells = totalAvailableCells + 1
                #Here is where we could generate a map that stores the cells for reduction.
                elif pixelValues[i][j] == "0":
                    totalAvailableCells = totalAvailableCells + 1
                    #Here is where we can create the map by checking the neighbors
                    if validPixelToChange(i, j, "increase") == 1:
# Then add the pixel as a potential pixel to change
mapAdditional.append(str(i) + "-" + str(j))  # append to the map, the current row and column

# This searches through the list and the recreates the map with only unique entries
makeUnique()

#for i in range(len(mapAdditional)):
    # j = mapAdditional[i].split('-')
    # GP.AddMessage("Row: " + j[0] + ", Col: " + j[1])
    # GP.AddMessage("Cell is: " + str(pixelValues[int(j[0])][int(j[1])]))

# Raster Properties (number of cells with 1, total number of cells)
rasProp = [totalCount.totalAvailableCells]

# return the total count
return rasProp

# Verifies if the pixel selected is valid for modification
# validity is determined by a cell touching at lest two cells in either direction.
# At least one cell must be directly above, below, left, or right.
# Get a valid cell to change
def validPixelToChange(Row, Col, changeType):

    # The valid value = 1 because that means adjacent cells have values.
    # The reason for now allowing valid to be 0 is because we don't want to have any randomly
    # placed seed to be
    # the center of reduction or addition in an area without a cut zone.
    valid = "1"
    changeable = 0

    try:
        # This will only allow for an increase in the case of a cell above, below, left, or right and its
        # adjacent to be == 1
        if (pixelValues[Row + 1][Col] == valid) and ((pixelValues[Row + 1][Col - 1] == valid) or
            (pixelValues[Row + 1][Col + 1] == valid)): changeable = 1  # Above and (one adjacent)
        if (pixelValues[Row - 1][Col] == valid) and ((pixelValues[Row - 1][Col - 1] == valid) or
            (pixelValues[Row - 1][Col + 1] == valid)): changeable = 1  # Below and (adjacent)
        if (pixelValues[Row][Col - 1] == valid) and ((pixelValues[Row + 1][Col - 1] == valid) or
            (pixelValues[Row - 1][Col - 1] == valid)): changeable = 1  # Left and (adjacent)
        if (pixelValues[Row][Col + 1] == valid) and ((pixelValues[Row + 1][Col + 1] == valid) or
            (pixelValues[Row - 1][Col + 1] == valid)): changeable = 1  # Right and (adjacent)

        # XX or X
        # X  XX
        if (pixelValues[Row][Col + 1] == valid) and ((pixelValues[Row + 1][Col] == valid) or
            (pixelValues[Row - 1][Col] == valid)): changeable = 1  # right and (above or below)

        # XX or X
        # X  XX
        if (pixelValues[Row][Col - 1] == valid) and ((pixelValues[Row + 1][Col] == valid) or
            (pixelValues[Row - 1][Col] == valid)): changeable = 1  # left and (above or below)

        # if it did not meet any of the criteria then, default
        if changeable == 1:
            GP.AddMessage("This pixel can be changed")
return 1
else:
    #GP.AddMessage("This pixel can NOT be changed")
    return 0

except:
    GP.AddMessage("Failed to validate the pixel")
    #This likely happened because the selected scale went outside of the bounds of the image.
    #The default buffer is not enough to cover the assessment. Basically if a cell is on the
    #border, then the buffer only allows for one cell to the outside, but when validating the pixel it needs
to check the
    #neighbor of the neighbor. For Instance, if the top cell is being checked,
    #the current algorithm verify's the cell above, then the validate the cell above that, going
beyond the buffer.
    return 0

def makeUnique():

    values=[]
    for v in range(len(mapAdditional)):
        if not mapAdditional[v] in values:
            #GP.AddMessage("Unique Value")
            values.append(mapAdditional[v])

    for j in range(len(mapAdditional)):
        mapAdditional.pop()  #removed all the values from mapAdditional

    for i in range(len(values)):
        mapAdditional.append(values[i])

    #Change a cells value so the total count remains correct
    def modifyValue(changeAmount):

        #This array stores the same values as the other arrays, but most importantly it stores the
        #potential locations for changing the value. This will only be used for decreasing problems.
        #Each time a random location is looked at, if that location cannot be changed, then it will get rid
        #of
        #that location as a possibility. Therefore, the maximum number of iterations is the total number
        #times the number of cells
        #to be changed.
        while 1:

            if changeAmount == "increase":
                #GP.AddMessage("Length of mapAdditional: " + str(len(mapAdditional)))

                #ensures that we are looking at a unique list of optional pixels to change
                #makeUnique()

                locationString = mapAdditional.pop(random.randrange(0, len(mapAdditional)))  #Gets a
row and column from the available map

    return 1
randString = locationString.split('-')

# this takes the string and creates an integer value from it. I only did this because I already have the if statements written
randLoc = []
randLoc.append(int(randString[0]))
randLoc.append(int(randString[1]))

if pixelValues[randLoc[0]][randLoc[1]] == "0":
    #GP.AddMessage("Increasing amount")
    pixelValues[randLoc[0]][randLoc[1]] = "1"

    # now check if that effects each neighboring cell, if that neighboring cell can be modified, add it to the list
    # but only check if the neighboring cell is 0, because if it is already 1 we can't add to it!

    if pixelValues[randLoc[0]+1][randLoc[1]-1] == "0": # Above(Left)
        if validPixelToChange(randLoc[0]+1,randLoc[1]-1,"increase") == 1:
            mapAdditional.append(str(randLoc[0]+1) + "-" + str(randLoc[1]-1)) # append to the map, the current row and column

    if pixelValues[randLoc[0]+1][randLoc[1]+1] == "0": # Above(Right)
        if validPixelToChange(randLoc[0]+1,randLoc[1]+1,"increase") == 1:
            mapAdditional.append(str(randLoc[0]+1) + "-" + str(randLoc[1]+1)) # append to the map, the current row and column

    if pixelValues[randLoc[0]+1][randLoc[1]] == "0": # Above
        if validPixelToChange(randLoc[0]+1,randLoc[1],"increase") == 1:
            mapAdditional.append(str(randLoc[0]+1) + "-" + str(randLoc[1])) # append to the map, the current row and column

    if pixelValues[randLoc[0]][randLoc[1]+1] == "0": # Right
        if validPixelToChange(randLoc[0],randLoc[1]+1,"increase") == 1:
            mapAdditional.append(str(randLoc[0]) + "-" + str(randLoc[1]+1)) # append to the map, the current row and column

    if pixelValues[randLoc[0]-1][randLoc[1]-1] == "0": # Below(Left)
        if validPixelToChange(randLoc[0]-1,randLoc[1]-1,"increase") == 1:
            mapAdditional.append(str(randLoc[0]-1) + "-" + str(randLoc[1]-1)) # append to the map, the current row and column

    if pixelValues[randLoc[0]-1][randLoc[1]] == "0": # Below
        if validPixelToChange(randLoc[0]-1,randLoc[1],"increase") == 1:
            mapAdditional.append(str(randLoc[0]-1) + "-" + str(randLoc[1])) # append to the map, the current row and column

    if pixelValues[randLoc[0]-1][randLoc[1]+1] == "0": # Below(Right)
        if validPixelToChange(randLoc[0]-1,randLoc[1]+1,"increase") == 1:
            mapAdditional.append(str(randLoc[0]-1) + "-" + str(randLoc[1]+1)) # append to the map, the current row and column
break
else:
    # Generate Random Number to select Row
    randRow = random.randrange(1, len(pixelValues) - 1)
    # Generate Random Number to select Column
    randCol = random.randrange(1, len(pixelValues[randRow]) - 1)

    # Verify that the pixel is currently = to 1
    if pixelValues[randRow][randCol] == "1":
        # Make a counter that keeps track of the number of corners that are intact
        cornerCounter = 0
        sideCounter = 0

        # First check to eliminate 2 cases where there are 7 or more surrounding 1s
        # As long as there are more than 1 side and less than four, continue to the next step

        # Check corner to see if it is within a cut
        if pixelValues[randRow + 1][randCol - 1] == "1": # Above(Left)
            cornerCounter = cornerCounter + 1
        if pixelValues[randRow + 1][randCol + 1] == "1": # Above(Right)
            cornerCounter = cornerCounter + 1
        if pixelValues[randRow - 1][randCol - 1] == "1": # Below(Left)
            cornerCounter = cornerCounter + 1
        if pixelValues[randRow - 1][randCol + 1] == "1": # Below(Right)
            cornerCounter = cornerCounter + 1

        # Check all sides to see how many exist
        if pixelValues[randRow + 1][randCol] == "1": # Above
            sideCounter = sideCounter + 1
        if pixelValues[randRow - 1][randCol] == "1": # Below
            sideCounter = sideCounter + 1
        if pixelValues[randRow][randCol + 1] == "1": # Right
            sideCounter = sideCounter + 1
        if pixelValues[randRow][randCol - 1] == "1": # Left
            sideCounter = sideCounter + 1

        if (2 <= sideCounter < 4): # less than all sides are not equal to 1
            # Set the current cell to 0 so we can check if it would effect the neighboring cells
            pixelValues[randRow][randCol] = "0"

            # Set a temporary variable that stores whether or not any of the side cells do not
            # meet adjacency rules
            restricted = 0

            # For each neighboring cell verify that it meets the adjacency (continuity) requirement

            # Because we know that at least 1 but not more than 3 cells bordering this one,
            # then let's check all of the sides and if they are =1 to one, make sure that adjusting
            # the current cell
            # wouldn't violate any of the side cell's adjacency requirements.

            if pixelValues[randRow+1][randCol] == "1": # Above
                if validPixelToChange(randRow+1, randCol, "decrease") == 0:
restricted = 1

if pixelValues[randRow+1][randCol-1] == "1": #Above(Left)
    if validPixelToChange(randRow+1, randCol-1, "decrease") == 0:
        restricted = 1

if pixelValues[randRow+1][randCol+1] == "1": #Above(Right)
    if validPixelToChange(randRow+1, randCol+1, "decrease") == 0:
        restricted = 1

if pixelValues[randRow][randCol-1] == "1": #Left
    if validPixelToChange(randRow, randCol-1, "decrease") == 0:
        restricted = 1

if pixelValues[randRow][randCol+1] == "1": #Right
    if validPixelToChange(randRow, randCol+1, "decrease") == 0:
        restricted = 1

elif pixelValues[randRow-1][randCol-1] == "1": #Below(Left)
    if validPixelToChange(randRow-1, randCol-1, "decrease") == 0:
        restricted = 1

if pixelValues[randRow-1][randCol] == "1": #Below
    if validPixelToChange(randRow-1, randCol, "decrease") == 0:
        restricted = 1

elif pixelValues[randRow-1][randCol+1] == "1": #Below(Right)
    if validPixelToChange(randRow-1, randCol+1, "decrease") == 0:
        restricted = 1

if restricted == 0:
    #GP.AddMessage("########################################################################### Changed 
    #GP.AddMessage("###########################################################################")
    break
else:
    #Changing this pixel would break the adjacency requirements for another pixel
    #This pixel should not be changed, to set it back to 1 and try again
    pixelValues[randRow][randCol] = "1"
    continue
else:
    #This pixel is not on the border
    continue
    GP.AddMessage("This cell is NOT allowable")
else:
    #This pixel is not == to "1"
    #look for another possible option
    continue

#Write a new file
def writeNewFile(outText):
    #then when writing the raster back to a binary, do not output the buffer values around the raster...
    #looping variable
    s = 0
    i = 0
    #Prepare new file to write to
    outFile = open(outText, 'w')
#write the header information
for s in range(len(headerInfo)):
    outFile.write(headerInfo[s])

#write new values into outText file
for i in range(1, len(pixelValues) -1):
    for j in range(1, len(pixelValues[i]) -1): #This starts at 1 and does not read the final value in
        the row, effectively eliminating the buffer on each side.
        outFile.write(pixelValues[i][j]) #write the value of the cell (or header info)
        if j+1 < len(pixelValues[i]):
            outFile.write(" ") #write a space
        outFile.write("\n")

#Close the outFile because we are done writing to it
outFile.close()

GP.AddMessage("Finished Writing")

try:

####################################################
# Input Files
####################################################

#The ASCII file to read from
Input_Harvest_Plan = sys.argv[1]

#The output file
Output_Modified_Harvest_Plan = sys.argv[2]

#The output number of visible cells after it has been cleaned
#zero is the default
Number_of_Cells_Not_Visible = 0

#Other variables important variables
headerInfo = [] #temporarily stores the header information
pixelValues = [] #temporarily stores each pixel value
mapAdditional = [] #temporarily stores a map of cells available to modify in an increasing
direction

#read the ASCII file into memory
readInValues(Input_Harvest_Plan)

#Clean the solution so all the pixels match the adjacency requirement
cleanChild()

#Running twice to ensure that all pixels are removed properly
#This could go away if I enable the additional verifyPixelValid statements for right and (up
down) or left (up down)
cleanChild()

#get the total count of all cells = 1 in the raster (ASCII)
rasterProperties = get count()

#GP.AddMessage("Number planned for harvest: " + str(int(rasterProperties[0])))
writeNewFile(Output_Modified_Harvest_Plan)

#This returns the total number of cells that are not visible.
#Take the number of required harvest cells and subtract these cells to derive a new amount.

Number_of_Cells_Not_Visible = rasterProperties[0]
#GP.AddMessage(Number_of_Cells_Not_Visible)
#GP.SetParameterAsText(3, "10")

eexcept:
    GP.AddMessage("Error: Could not complete the modification\n")
Appendix 6

The information below uses the format described Table 3.1.

Appendix 6.1 Hypothetical Scenarios

Appendix 6.1.1 Scenario #1: Flat

<table>
<thead>
<tr>
<th>Required Timber Harvest</th>
<th>80%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Extraction Level</td>
<td></td>
</tr>
<tr>
<td>Desired AVA</td>
<td>1%</td>
</tr>
<tr>
<td>Viewing Height</td>
<td>1.5m</td>
</tr>
<tr>
<td>Invisible Cells</td>
<td></td>
</tr>
</tbody>
</table>

| Initial Parent AVA      | ParentA: 9.12% |
|                        | ParentB: 59.84%|
| Number of generations   | 17           |
| until completion        |              |
| Type of completion      | Automatic     |
| Best Percent Visible    | .48%         |
| Alteration found        |              |
| Best harvest solution   |              |

![Figure A6.1 Scenario #1 1.5m Vis Invisible Cells](image1)

![Figure A6.2 Scenario #1 1.5m Best Solution](image2)
Visible cells after harvest of best solution

Figure A6.3 Scenario #1 1.5m Visible Cells

Output log data

<table>
<thead>
<tr>
<th>Percent Visible Alteration</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
</tr>
<tr>
<td>80</td>
</tr>
<tr>
<td>60</td>
</tr>
<tr>
<td>40</td>
</tr>
<tr>
<td>20</td>
</tr>
<tr>
<td>0</td>
</tr>
</tbody>
</table>

Generation

Figure A6.4 Scenario #1 Ground Vis Solution Progression

Required Timber Harvest 80%
Extraction Level
Desired AVA 1%
Viewing Height 41.5m
<table>
<thead>
<tr>
<th><strong>Invisible Cells</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Figure A6.5 Scenario #2 41.5m Vis Invisible Cells</strong></td>
</tr>
<tr>
<td><strong>Initial Parent AVA</strong></td>
</tr>
<tr>
<td>ParentA: 19.52%</td>
</tr>
<tr>
<td>ParentB: 57.92%</td>
</tr>
<tr>
<td><strong>Number of generations until completion</strong></td>
</tr>
<tr>
<td>17</td>
</tr>
<tr>
<td><strong>Type of completion</strong></td>
</tr>
<tr>
<td>Automatic</td>
</tr>
<tr>
<td><strong>Best Percent Visible Alteration found</strong></td>
</tr>
<tr>
<td>.8%</td>
</tr>
<tr>
<td><strong>Best harvest solution found</strong></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td><strong>Visible cells after harvest of best solution</strong></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td><strong>Figure A6.6 Scenario #1 41.5m Best Solution</strong></td>
</tr>
<tr>
<td><strong>Figure A6.7 Scenario #1 41.5m Visible Cells</strong></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>
Figure A6.8 Scenario #1 Med Vis Solution Progression

Appendix 6.1.2 Scenario #2: Four Medium Hills

| Required Timber Harvest Extraction Level | 80%          |
| Desired AVA                             | 1%           |
| Viewing Height                          | 1.5m         |
| Invisible Cells                         |              |

![Figure A6.9 Scenario #2 1.5m Vis Invisible Cells]

Initial Parent AVA
- ParentA: 2.88%
- ParentB: 1.28%

Number of generations until completion
- 3

Type of completion
- Automatic

Best Percent Visible
- .96%
<table>
<thead>
<tr>
<th>Alteration found</th>
</tr>
</thead>
<tbody>
<tr>
<td>Best harvest solution found</td>
</tr>
</tbody>
</table>

![Figure A6.10 Scenario #2 1.5m Best Solution](image1)

<table>
<thead>
<tr>
<th>Visible cells after harvest of best solution</th>
</tr>
</thead>
</table>

![Figure A6.11 Scenario #2 1.5m Visible Cells](image2)

<table>
<thead>
<tr>
<th>Output log data</th>
</tr>
</thead>
</table>


### Scenario #2 Ground Vis Solution Progression

| Required Timber Harvest Extraction Level | 80% |
| Desired AVA | 1% |
| Viewing Height | 41.5m |

#### Invisible Cells

- **Initial Parent AVA**: ParentA: 7.52%, ParentB: 4.32%
- **Number of generations until completion**: 22
- **Type of completion**: Automatic

---

*Figure A6.12 Scenario #2 Ground Vis Solution Progression*

*Figure A6.13 Scenario #2 41.5m Vis Invisible Cells*
Best Percent Visible Alteration found

.96%

Best harvest solution found

Visible cells after harvest of best solution

Output log data

Figure A6.14 Scenario #2 41.5m Best Solution

Figure A6.15 Scenario #2 41.5m Visible Cells
### Appendix 6.1.3 Scenario #3: High Mountain Face

| Required Timber Harvest Extraction Level | 80%          |
| Desired AVA                              | 1%           |
| Viewing Height                           | 1.5m         |
| Invisible Cells                          |              |

**Initial Parent AVA**
- ParentA: 64.48%
- ParentB: 51.2%

<p>| Number of generations until completion   | 207          |
| Type of completion                      | Manual       |
| Best Percent Visible                    | 28.96%       |</p>
<table>
<thead>
<tr>
<th>Alteration found</th>
<th><img src="image" alt="Figure A6.18 Scenario #3 1.5m Best Solution" /></th>
</tr>
</thead>
<tbody>
<tr>
<td>Best harvest solution found</td>
<td><img src="image" alt="Figure A6.19 Scenario #3 1.5m Visible Cells" /></td>
</tr>
<tr>
<td>Visible cells after harvest of best solution</td>
<td></td>
</tr>
<tr>
<td>Output log data</td>
<td></td>
</tr>
</tbody>
</table>
Required Timber Harvest Extraction Level 80%

Desired AVA 1%

Viewing Height 41.5m

Invisible Cells

Initial Parent AVA ParentA: 68.8%
ParentB: 61.12%

Number of generations until completion 195

Type of completion Manual
<table>
<thead>
<tr>
<th>Best Percent Visible Alteration found</th>
<th>32.96%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Best harvest solution found</td>
<td><img src="image1.png" alt="Image" /> Figure A6.22 Scenario #3 41.5m Best Solution</td>
</tr>
<tr>
<td>Visible cells after harvest of best solution</td>
<td><img src="image2.png" alt="Image" /> Figure A6.23 Scenario #3 41.5m Visible Cells</td>
</tr>
<tr>
<td>Output log data</td>
<td></td>
</tr>
</tbody>
</table>
Figure A6.24 Solution #3 Med Vis Solution Progression

Required Timber Harvest
Extraction Level 60%

Desired AVA 1%

Viewing Height 41.5m

Invisible Cells

Figure A6.25 Scenario #3 41.5m Vis 60% Timber Invisible Cells

Initial Parent AVA
ParentA: 30.24%
ParentB: 40.16%

Number of generations until completion 170

Type of completion Manual
| Best Percent Visible Alteration found | 11.36% |
| Best harvest solution found | ![Image](#) |
| Visible cells after harvest of best solution | ![Image](#) |
| Output log data | ![Image](#) |
Required Timber Harvest Extraction Level 80%
Resolution 20m, 75x75 grid
Desired AVA 1%
Viewing Height 41.5m
Invisible Cells

Figure A6.28 Solution #3 Med Vis 60% Solution Progression

Initial Parent AVA ParentA: 67.47%
ParentB: 57.92%
Number of generations until completion 146

Figure A6.29 Scenario #3 41.5m Vis 60% Timber Invisible Cells
<table>
<thead>
<tr>
<th>Type of completion</th>
<th>Manual</th>
</tr>
</thead>
<tbody>
<tr>
<td>Best Percent Visible</td>
<td>13.65%</td>
</tr>
<tr>
<td>Alteration found</td>
<td></td>
</tr>
<tr>
<td>Best harvest solution</td>
<td></td>
</tr>
<tr>
<td>found</td>
<td></td>
</tr>
</tbody>
</table>

Figure A6.30 Scenario #3 41.5m 60% Timber Best Solution

Visible cells after harvest of best solution

Figure A6.31 Scenario #3 41.5m 60% Timber Visible Cells

Output log data
Appendix 6.1.4 Scenario #4: One High Hill

Required Timber Harvest 80%
Extraction Level

Desired AVA 1%
Viewing Height 1.5m

Invisible Cells

Initial Parent AVA
ParentA: 5.92%
ParentB: 13.28%

Number of generations until completion 21
Type of completion | Automatic
---|---
Best Percent Visible Alteration found | .64%
Best harvest solution found | 

![Figure A6.34 Scenario #4 1.5m Best Solution](image1)

Visible cells after harvest of best solution |

![Figure A6.35 Scenario #4 1.5m Visible Cells](image2)

Output log data
**Figure A6.36 Scenario #4 Ground Vis Solution Progression**

| Required Timber Harvest Extraction Level | 80% |
| Desired AVA                               | 1%  |
| Viewing Height                            | 41.5m |
| Invisible Cells                           | ![Blank] |

**Figure A6.37 Scenario #4 41.5m Vis Invisible Cells**

| Initial Parent AVA                        | ParentA: 16.64%  
<p>|                                         | ParentB: 18.88%  |
| Number of generations until completion   | 239             |
| Type of completion                       | Manual          |</p>
<table>
<thead>
<tr>
<th>Best Percent Visible Alteration found</th>
<th>1.44%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Best harvest solution found</td>
<td><img src="image" alt="Figure A6.38 Scenario #4 41.5m Best Solution" /></td>
</tr>
<tr>
<td>Visible cells after harvest of best solution</td>
<td><img src="image" alt="Figure A6.39 Scenario #4 41.5m Visible Cells" /></td>
</tr>
<tr>
<td>Output log data</td>
<td></td>
</tr>
</tbody>
</table>
Appendix 6.1.5 Scenario #5: One High Hill and Three Medium Hills

| Required Timber Harvest Extraction Level | 80% |
| Desired AVA                              | 1%  |
| Viewing Height                           | 1.5m |
| Initial Parent AVA                       |     |
| ParentA: 0.16%                           |     |
| ParentB: 5.44%                           |     |

Invisible Cells

Number of generations until completion: 0
Type of completion: Automatic
Best Percent Visible: 0.16%
| Alteration found |  
| Best harvest solution found |  
| Visible cells after harvest of best solution |  

Figure A6.42 Scenario #5 1.5m Best Solution

Figure A6.43 Scenario #5 1.5m Visible Cells

| Required Timber Harvest Extraction Level | 80% |
| Desired AVA | 1% |
| Viewing Height | 41.5m |
| Initial Parent AVA | ParentA: 2.24% ParentB: 6.24% |

| Invisible Cells |  

Figure A6.44 Scenario #5 41.5m Vis Invisible Cells
<table>
<thead>
<tr>
<th>Number of generations until completion</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type of completion</td>
<td>Automatic</td>
</tr>
<tr>
<td>Best Percent Visible Alteration found</td>
<td>.96%</td>
</tr>
<tr>
<td>Best harvest solution found</td>
<td><img src="image" alt="Figure A6.45 Scenario #5 41.5m Best Solution" /></td>
</tr>
<tr>
<td>Visible cells after harvest of best solution</td>
<td><img src="image" alt="Figure A6.46 Scenario #5 41.5m Visible Cells" /></td>
</tr>
<tr>
<td>Output log data</td>
<td>171</td>
</tr>
</tbody>
</table>
Figure A6.47 Scenario #4 Med Vis Solution Progression

Figure A6.48 One High Hill Three Small Hills render at 41.5m view point

Figure A6.49 One High Hill Three Small Hills render at 1241.5m view point

Appendix 6.1.6 Scenario #6: Two High Hills

<table>
<thead>
<tr>
<th>Required Timber Harvest</th>
<th>Extraction Level</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>80%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Desired AVA</th>
<th>Viewing Height</th>
</tr>
</thead>
<tbody>
<tr>
<td>1%</td>
<td>1.5m</td>
</tr>
</tbody>
</table>
| Initial Parent AVA                | ParentA: 8.8%  
                      | ParentB: 12.96% |
|----------------------------------|----------------|
| Invisible Cells                  | **Figure A6.50 Scenario #6 1.5m Vis Invisible Cells** |
| Number of generations until completion | 116            |
| Type of completion               | Automatic      |
| Best Percent Visible             | .64%           |
| Alteration found                 |                |
| Best harvest solution found      |                |
| Visible cells after harvest of best solution | **Figure A6.51 Scenario #6 1.5m Best Solution** |
| Output log data                  |                |
### Scenario #6 Ground Vis Solution Progression

**Required Timber Harvest**
- Extraction Level: 95%

**Desired AVA**
- 1%

**Viewing Height**
- 41.5m

**Initial Parent AVA**
- ParentA: 33.6%
- ParentB: 37.6%

**Invisible Cells**

<table>
<thead>
<tr>
<th>Generation</th>
<th>Percent Visible Alteration</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
</tr>
<tr>
<td>51</td>
<td></td>
</tr>
<tr>
<td>101</td>
<td></td>
</tr>
</tbody>
</table>

**Figure A6.53 Scenario #6 Ground Vis Solution Progression**

**Figure A6.54 Scenario #6 41.5m Vis 95% Timber Invisible Cells**

**Number of generations until completion**
- 451

**Type of completion**
- Manual
<table>
<thead>
<tr>
<th>Best Percent Visible Alteration found</th>
<th>17.92%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Best harvest solution found</td>
<td></td>
</tr>
<tr>
<td>Visible cells after harvest of best solution</td>
<td></td>
</tr>
<tr>
<td>Output log data</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Figure A6.55 Scenario #6 41.5m 95% Timber Best Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure A6.56 Scenario #6 41.5m 95% Timber Visible Cells</td>
</tr>
</tbody>
</table>

175
Figure A6.57 Scenario #6 Med Vis 95% Solution Progression

| Required Timber Harvest Extraction Level | 90% |
| Desired AVA | 1% |
| Viewing Height | 41.5m |
| Initial Parent AVA | ParentA: 20.8%  
ParentB: 24.64% |

Invisible Cells

Figure A6.58 Scenario #6 41.5m Vis 90% Timber Invisible Cells

Number of generations until completion | 280 |
<table>
<thead>
<tr>
<th>Type of completion</th>
<th>Manual</th>
</tr>
</thead>
<tbody>
<tr>
<td>Best Percent Visible Alteration found</td>
<td>7.84%</td>
</tr>
<tr>
<td>Best harvest solution found</td>
<td></td>
</tr>
</tbody>
</table>

**Figure A6.59 Scenario #6 41.5m 90% Timber Best Solution**

- Visible cells after harvest of best solution

**Figure A6.60 Scenario #6 41.5m 90% Timber Visible Cells**

Output log data
**Figure A6.61 Scenario #6 Med Vis 90% Solution Progression**

| Required Timber Harvest Extraction Level | 85% |
| Desired AVA                             | 1%  |
| Viewing Height                          | 41.5m |
| Initial Parent AVA                      | ParentA: 14.24%  
|                                         | ParentB: 19.52%  |

**Figure A6.62 Scenario #6 41.5m Vis 85% Timber Invisible Cells**

<p>| Number of generations until completion | 236 |
| Type of completion                    | Manual |</p>
<table>
<thead>
<tr>
<th>Best Percent Visible Alteration found</th>
<th>3.36%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Best harvest solution found</td>
<td>![Image](Figure A6.63 Scenario #6 41.5m 85% Timber Best Solution)</td>
</tr>
<tr>
<td>Visible cells after harvest of best solution</td>
<td>![Image](Figure A6.64 Scenario #6 41.5m 85% Timber Visible Cells)</td>
</tr>
<tr>
<td>Output log data</td>
<td></td>
</tr>
</tbody>
</table>
### Figure A6.65 Scenario #6 Med Vis 85% Solution Progression

| Required Timber Harvest Extraction Level | 80% |
| Desired AVA | 1% |
| Viewing Height | 41.5m |
| Initial Parent AVA | ParentA: 16.48% |
| | ParentB: 16.64% |

#### Figure A6.66 Scenario #6 41.5m Vis 80% Timber Invisible Cells

<p>| Number of generations until completion | 245 |
| Type of completion | Manual |</p>
<table>
<thead>
<tr>
<th>Best Percent Visible Alteration found</th>
<th>1.92%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Best harvest solution found</td>
<td><img src="image" alt="Figure A6.67 Scenario #6 41.5m 80% Timber Best Solution" /></td>
</tr>
<tr>
<td>Visible cells after harvest of best solution</td>
<td><img src="image" alt="Figure A6.68 Scenario #6 41.5m 80% Timber Visible Cells" /></td>
</tr>
<tr>
<td>Output log data</td>
<td></td>
</tr>
</tbody>
</table>
Figure A6.69 Scenario #6 Med Vis 80% Solution Progression

Figure A6.70 Two High Hill render at 41.5m viewpoint

Figure A6.71 Two High Hill render at 1241.5m viewpoint

Required Timber Harvest 75%
Extraction Level
Desired AVA 1%
Viewing Height 41.5m
<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Initial Parent AVA</strong></td>
<td><strong>ParentA:</strong> 9.12%</td>
<td><strong>ParentB:</strong> 12.96%</td>
</tr>
<tr>
<td><strong>Invisible Cells</strong></td>
<td><img src="image1.png" alt="Image" /></td>
<td><img src="image2.png" alt="Image" /></td>
</tr>
<tr>
<td><strong>Figure A6.72 Scenario #6 41.5m Vis 75% Timber Invisible Cells</strong></td>
<td><img src="image3.png" alt="Image" /></td>
<td><img src="image4.png" alt="Image" /></td>
</tr>
<tr>
<td><strong>Number of generations until completion</strong></td>
<td>143</td>
<td></td>
</tr>
<tr>
<td><strong>Type of completion</strong></td>
<td>Automatic</td>
<td></td>
</tr>
<tr>
<td><strong>Best Percent Visible Alteration found</strong></td>
<td>.96%</td>
<td></td>
</tr>
<tr>
<td><strong>Best harvest solution found</strong></td>
<td><img src="image5.png" alt="Image" /></td>
<td><img src="image6.png" alt="Image" /></td>
</tr>
<tr>
<td><strong>Visible cells after harvest of best solution</strong></td>
<td><img src="image7.png" alt="Image" /></td>
<td><img src="image8.png" alt="Image" /></td>
</tr>
<tr>
<td><strong>Figure A6.73 Scenario #6 41.5m 75% Timber Best Solution</strong></td>
<td><img src="image9.png" alt="Image" /></td>
<td><img src="image10.png" alt="Image" /></td>
</tr>
<tr>
<td><strong>Figure A6.74 Scenario #6 41.5m 75% Timber Visible Cells</strong></td>
<td><img src="image11.png" alt="Image" /></td>
<td><img src="image12.png" alt="Image" /></td>
</tr>
<tr>
<td><strong>Output log data</strong></td>
<td></td>
<td><strong>183</strong></td>
</tr>
</tbody>
</table>
Figure A6.75 Scenario #6 Med Vis 75% Solution Progression

<table>
<thead>
<tr>
<th>Required Timber Harvest Extraction Level</th>
<th>70%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Desired AVA</td>
<td>1%</td>
</tr>
<tr>
<td>Viewing Height</td>
<td>41.5m</td>
</tr>
<tr>
<td>Initial Parent AVA</td>
<td>ParentA: 7.2%</td>
</tr>
<tr>
<td></td>
<td>ParentB: 11.36%</td>
</tr>
<tr>
<td>Invisible Cells</td>
<td></td>
</tr>
</tbody>
</table>

Figure A6.76 Scenario #6 41.5m Vis 70% Timber Invisible Cells

<table>
<thead>
<tr>
<th>Number of generations until completion</th>
<th>53</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type of completion</td>
<td>Automatic</td>
</tr>
</tbody>
</table>

184
<table>
<thead>
<tr>
<th>Best Percent Visible Alteration found</th>
<th>.96%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Best harvest solution found</td>
<td><img src="image" alt="Figure A6.77 Scenario #6 41.5m 70% Timber Best Solution" /></td>
</tr>
<tr>
<td>Visible cells after harvest of best solution</td>
<td><img src="image" alt="Figure A6.78 Scenario #6 41.5m 70% Timber Visible Cells" /></td>
</tr>
<tr>
<td>Output log data</td>
<td></td>
</tr>
<tr>
<td>Required Timber Harvest</td>
<td>65%</td>
</tr>
<tr>
<td>-------------------------</td>
<td>-----</td>
</tr>
<tr>
<td>Extraction Level</td>
<td></td>
</tr>
<tr>
<td>Desired AVA</td>
<td>1%</td>
</tr>
<tr>
<td>Viewing Height</td>
<td>41.5m</td>
</tr>
</tbody>
</table>
| Initial Parent AVA      | ParentA: 4.46%  
|                         | ParentB: 8.8%  |
| Invisible Cells         |     |

Number of generations until completion | 24
Type of completion                  | Automatic

Figure A6.79 Scenario #6 Med Vis 70% Solution Progression

Figure A6.80 Scenario #6 41.5m Vis 65% Timber Invisible Cells
Best Percent Visible Alteration found: .96%

Best harvest solution found:

Visible cells after harvest of best solution:

Output log data:
Figure A6.83 Scenario #6 Med Vis 65% Solution Progression

| Required Timber Harvest Extraction Level | 60% |
| Desired AVA | 1% |
| Viewing Height | 41.5 |
| Initial Parent AVA | ParentA: 2.88% |
| | ParentB: 6.4% |
| Invisible Cells | |

Figure A6.84 Scenario #6 41.5m Vis 60% Timber Invisible Cells

<p>| Number of generations until completion | 13 |
| Type of completion | Automatic |</p>
<table>
<thead>
<tr>
<th>Best Percent Visible Alteration found</th>
<th>.64%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Best harvest solution found</td>
<td><img src="image1" alt="Figure A6.85 Scenario #6 41.5m 60% Timber Best Solution" /></td>
</tr>
<tr>
<td>Visible cells after harvest of best solution</td>
<td><img src="image2" alt="Figure A6.86 Scenario #6 41.5m 60% Timber Visible Cells" /></td>
</tr>
<tr>
<td>Output log data</td>
<td></td>
</tr>
</tbody>
</table>
**Figure A6.87 Scenario #6 Med Vis 60% Solution Progression**

<table>
<thead>
<tr>
<th>Required Timber Harvest</th>
<th>55%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Extraction Level</td>
<td></td>
</tr>
<tr>
<td>Desired AVA</td>
<td>1%</td>
</tr>
<tr>
<td>Viewing Height</td>
<td>41.5M</td>
</tr>
</tbody>
</table>
| Initial Parent AVA      | ParentA: 1.6%  
                          | ParentB: 3.04% |
| Invisible Cells         |     |

**Figure A6.88 Scenario #6 41.5m Vis 55% Timber Invisible Cells**

<table>
<thead>
<tr>
<th>Number of generations until completion</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type of completion</td>
<td>Automatic</td>
</tr>
</tbody>
</table>
Best Percent Visible Alteration found 0.96%

Best harvest solution found

Visible cells after harvest of best solution

Required Timber Harvest Extraction Level 50%
Desired AVA 1%
Viewing Height 41.5m
Initial Parent AVA ParentA: 1.28%, ParentB: 0.64%
<table>
<thead>
<tr>
<th>Invisible Cells</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of generations until completion</td>
</tr>
<tr>
<td>Type of completion</td>
</tr>
<tr>
<td>Best Percent Visible Alteration found</td>
</tr>
<tr>
<td>Best harvest solution found</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Visible cells after harvest of best solution</th>
</tr>
</thead>
</table>

| Required Timber Harvest Extraction Level | 45% |

---

**Figure A6.91 Scenario #6 41.5m Vis 50% Timber Invisible Cells**

**Figure A6.92 Scenario #6 41.5m 55% Timber Best Solution**

**Figure A6.93 Scenario #6 41.5m 55% Timber Visible Cells**
<table>
<thead>
<tr>
<th>Desired AVA</th>
<th>1%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Viewing Height</td>
<td>41.5</td>
</tr>
</tbody>
</table>
| Initial Parent AVA         | ParentA: .16%  
<pre><code>                      | ParentB: 1.12% |
</code></pre>
<p>| Invisible Cells            | ![Image]  |
| number of generations until completion | 0        |
| Type of completion         | Automatic |
| Best Percent Visible Alteration found | .16      |
| Best harvest solution found| ![Image]  |
| Visible cells after harvest of best solution | ![Image]  |</p>
<table>
<thead>
<tr>
<th>Required Timber Harvest Extraction Level</th>
<th>40%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Desired AVA</td>
<td>1%</td>
</tr>
<tr>
<td>Viewing Height</td>
<td>41.5m</td>
</tr>
</tbody>
</table>
| Initial Parent AVA                     | ParentA: 0%  
ParentB: 0% |
| Invisible Cells                        | ![Image](image1.png) |
| Figure A6.97 Scenario #6 41.5m Vis 40% Timber Invisible Cells |
| Number of generations until completion | 0   |
| Type of completion                     | Automatic |
| Best Percent Visible Alteration found  | 0%  |
| Best harvest solution found            | ![Image](image2.png) |
| Figure A6.98 Scenario #6 41.5m 40% Timber Best Solution |
Visible cells after harvest of best solution

Figure A6.99 Scenario #6 41.5m 40% Timber Visible Cells

Appendix 6.1.7  Scenario #7: Two High Mountains with a Valley

| Required Timber Harvest Extraction Level | 80% |
| Desired AVA | 1% |
| Viewing Height | 1.5m |
| Initial Parent AVA | ParentA: 2.4%  
ParentB: 3.2% |
| Invisible Cells | |

Figure A6.100 Scenario #7 1.5m Vis Invisible Cells

<p>| Number of generations until completion | 5 |
| Type of completion | Automatic |
| Best Percent Visible Alteration found | .96% |</p>
<table>
<thead>
<tr>
<th>Best harvest solution found</th>
<th><img src="image1" alt="Figure A6.101 Scenario #7 1.5m Best Solution" /></th>
</tr>
</thead>
<tbody>
<tr>
<td>Visible cells after harvest of best solution</td>
<td><img src="image2" alt="Figure A6.102 Scenario #7 1.5m Visible Cells" /></td>
</tr>
<tr>
<td>Output log data</td>
<td></td>
</tr>
</tbody>
</table>

---

196
Figure A6.103 Scenario #7 Ground Vis Solution Progression

| Required Timber Harvest Extraction Level | 80% |
| Desired AVA | 1% |
| Viewing Height | 41.5m |
| Initial Parent AVA | ParentA: 4.16%  
ParentB: 3.84% |
| Invisible Cells |  
Figure A6.104 Scenario #7 41.5m Vis Invisible Cells |
| Number of generations until completion | 18 |
| Type of completion | Automatic |
| Best Percent Visible | .96% |
Alteration found

Best harvest solution found

Visible cells after harvest of best solution

Output log data
Figure A6.107 Scenario #7 Med Vis Solution Progression

Figure A6.108 High Mountain Valley 3D render at 41.5m view point

Figure A6.109 Two High Mountain Valley 3D render at 1241.5m view point

Appendix 6.2

Appendix 6.2.1 Tree Farm License 49, Kelowna, BC

<p>| Required Timber Harvest Extraction Level | 80% |
| Desired AVA                            | 1%  |</p>
<table>
<thead>
<tr>
<th>Viewing Height</th>
<th>41.5m</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial Parent AVA</td>
<td>ParentA: 7.64%</td>
</tr>
<tr>
<td></td>
<td>ParentB: 7.62%</td>
</tr>
<tr>
<td>Invisible Cells</td>
<td></td>
</tr>
</tbody>
</table>

| Number of generations until completion | 119           |
| Type of completion                     | Manual        |
| Best Percent Visible Alteration found  | 6.77%         |

**Figure A6.110 Scenario #7 41.5m Vis Invisible Cells**
Best harvest solution found

Figure A6.111 Scenario #7 41.5m Best Solution
Visible cells after harvest of best solution

Figure A6.112 Scenario #7 41.5m Visible Cells

Output log data

Data is not given because initial and final values were too similar.

Figure A6.113 High Mountain Valley 3D render at 41.5m view point

Figure A6.114 Two High Mountain Valley 3D render at 1241.5m view point
Appendix 7

For those who have used ESRI's ArcInfo, they will be well aware of the power and wide array of tools available. So, the software may not seem like much of a limitation, but the amount of complexity, requirements and objectives for this thesis pushed the limits of this development solution. Though ArcInfo helped boost the speed of development, it must also be listed as a limitation because the program had major memory leaks, is quite inefficient and was lacking some key raster tools.

In terms of efficiency and processing time, a large portion of programming and modeling went into finding the most efficient path to a solution. Even with all the tweaks and fixes, one generation can take roughly ten minutes to complete. This is a major limitation because as the complexity rises, the GA can require more generations to solve. Consider a problem where the GA needed to create the most dispersed grid. On a single grid, 66% of the cells are required to be black and 33% are required to be white. A few potential possibilities that the GA might happen upon are shown on the 3x3 grid below:

![Possibility #1](image1)  ![Possibility #2](image2)  ![Possibility #3](image3)  ![Possibility #4](image4)  ![Possibility #5](image5)

Figure A7.1 Potential cell arrangement possibilities within Model

Possibility #5 appears to be the most dispersed, but are there other potential possibilities? The number can be calculated using combinatorial calculations: \( nCk = \frac{n!}{k!(n-k)!} \). In the case of a 3x3 grid, where 6 cells need to be black, this would mean that there are 84 possibilities. When increasing the complexity to a mere 4x4 grid, with 11 cells being selected as black, the number of combination possibilities jumps to 4368! What does this all mean? It means that if ArcInfo only processes one generation consisting of 10 possibilities in 10 minutes, it better be a smart model because time is of the essence.
Initially, ArcInfo seemed to have exactly the right tools and functionality required for this model. Yet, some problems with the thesis were encountered months before development because some of the key features ESRI professed were not available in version 9.1. It was not until the upgrade to 9.2 that a number of the required features were available without additional system-level scripting. However, midway through development, even the additional tools in the new model were not enough for the requirements. ESRI's online forums and telephone support service still did not offer any headway for a solution to some development problems, so additional tools had to be created using Python.
Appendix 8

The information in this section is taken verbatim from ArcGIS Desktop Help, under the Spatial Analyst, performing a viewshed analysis.

Viewshed identifies the cells in an input raster that can be seen from one or more observation points or lines. Each cell in the output raster receives a value that indicates how many observer points can be seen from each location. If you have only one observer point, each cell that can see that observer point is given a value of 1. All cells that cannot see the observer point are given a value of 0. The observer points feature class can contain points or lines. The nodes and vertices of lines will be used as observation points.

Why calculate viewshed?

Viewshed is useful when you want to know how visible objects might be—for example, From which locations on the landscape will the water towers be visible if they are placed in this location? or What will the view be from this road?

In the example below, the viewshed from an observation tower is identified. The elevation raster displays the height of the land (darker locations represent lower elevations), and the observation tower is marked as a green triangle. The height of the observation tower can be specified in the analysis. Cells in green are visible from the observation tower, while cells in red are not.
Displaying a hillshade underneath your elevation and the output from the Viewshed tool is a useful technique for visualizing the relationship between visibility and terrain.

Not only can you determine which cells can be seen from the observation tower, if you have several observation points, you can also determine which observers can see each observed location. Knowing which observer can see which locations can affect decision making. For example, in a visual quality study for siting a landfill, if it is determined that the proposed landfill can only be seen from dirt roads and not from the primary and secondary roads, it may be deemed a favorable location.

**Controlling the viewshed**

The image below graphically depicts how a viewshed is performed. The observation point is on the mountain top to the left (at OF1 in the image). The direction of the viewshed is within the cone looking to the right. You can control how much to offset the observation point (for example, the height of the tower), the direction to look, and how high and low to look from the horizon.
There are nine characteristics of the viewshed that you can control:

1. The surface elevations for the observation points (Spot)
2. The vertical distance in surface units to be added to the z-value of the observation points (OffsetA)
3. The vertical distance in surface units to add to the z-value of each cell as it is considered for visibility (OffsetB)
4. The start of the horizontal angle to limit the scan (Azimuth1)
5. The end of the horizontal angle to limit the scan (Azimuth2)
6. The top of the vertical angle to limit the scan (Vert1)
7. The bottom of the vertical angle to limit the scan (Vert2)
8. The inner radius that limits the search distance when identifying areas visible from each observation point (Radius1)
9. The outer radius that limits the search distance when identifying areas visible from each observation point (Radius2)
Appendix 9

This section explains how the perspective view, percent visible alteration was calculated. Two calculation methods were used in the thesis, although one was used within the Model and the other was used as a supplement to the Model's results. A brief description of the two methods can be found in Section 2.4.

The first method uses the viewshed analysis tool available in ArcInfo. For more information about the viewshed analysis tool please refer to Appendix 8. The first calculation method is a quick way to gauge a perspective view assessment but may not be the most precise. Precision is lost because the method is highly reliant upon the VSU and only returns if a cell is visible or not. The potential implications are that if the VSU extends into an area that normally would not be considered in a VSU (e.g., areas beyond a hill or mountain) then the results of percent visible alteration could be inflated. Furthermore, because the tool is only measuring whether or not the cell is visible, cells at very low angles received the same amount of weight of cells are high angles or high slopes. However, in a real world scenario it is unlikely an area with a low angle with be noticed as much as an area with a high angle. Figure A9.1 simplifies the viewshed analysis. In Figure A9.1 the black dot represents the center of three example cells. Because all three cells are visible, the will be counted with the same weight, although it is easy to understand that the cell at the highest slope would be more visible than the cell with the lowest slope.

![Figure A9.1 Viewshed Illustration](image)

To make the assessment more precise, a method that measures how much of the cell is visible, needed to be developed. ArcInfo contains a tool called hillshade that provides a way to measure the amount of light reflectance from a cell, and thereby offer a way to measure how much of the
cell is visible. Before proceeding in this section, please refer to Appendix 10 which provides a more thorough explanation of the *hillshade* tool. The tool allows one to measure the amount of light that reflects back to the light source (e.g. the sun). This method was fitted to the current problem by creating a situation where the light source was the elevation and location of the viewpoint. The tool requires an azimuth or angle direction of the sun and the altitude of the sun. So by modifying these two input parameters to make the viewpoint represent the sun, the amount of light reflectance could be accurately measured.

![Diagram](image)

**Figure A9.2 Using the azimuth within the hillshade tool**

Figure A9.2 depicts the DEM from Scenario #3, which has been made more opaque for illustration purposes. The arrows reflect the direction the viewer is looking at the landscape and also the direction of the sun as used in hillshade. The dark circle at the center of the crosshair is the viewpoint (distance not to scale) and theta is the angle that will be used in the azimuth. The angle was calculated as 225 degrees.
Figure A9.3 Using Angle within the hillshade tool (not to scale)

Figure A9.3 depicts the method to derive the angle of the sun from the horizon in order to simulate the sun at the height of the viewpoint. The diagram provided uses a gray line to depict the face of a steep slope. The 3,000 meter line is the distance from the center of the slope to the viewpoint. The 41.5 meter height is the height of the viewpoint. Theta is derived by getting the tangent of Theta. This becomes the angle of the sun.

Figure A9.2 and Figure A9.3 are used to produce a weighted raster in 2D that represents the amount of light that reflects back to the viewpoint. The outcome is a raster as shown in Figure A9.4. The raster produces value that range from 0 (black) to (255) white. A 0 value cell means that the cell is not visible and therefore will not be weighted in the percent visible alteration calculation in this method. To derive the final calculation for this method, the raster shown in Figure A9.4 is multiplied times the visUgly raster which identified pixels that are visible after the harvest. These two raster combined produce a weighted raster that identified how much of each visible cell is visible in perspective view shown in Figure A9.5. Finally, taking the weighted raster and dividing it by the highest potential light reflectance, which is 256 for every cell, a complete percent visible alteration (PVA) is produced.
As one can see, Figure A9.5 weighs the cells in the front of the slope much less than the cells at the high end of the slope. In order to verify the accuracy of these results a quick 3D render of Scenario #3 was done using Visual Nature Studio as shown in Figure A9.6. In the render, a tree was substituted as a simple white block and the surface was made to be gray. Then the calculation of the amount of visible surface (gray) divided by the amount of gray and white cells was compared to the total weighted raster from Figure A9.5. The difference between the two calculations was less than 1%, suggesting that the alternative method is very precise and compares well to a 3D render of the landscape after the harvest.
Figure A9.6 VNS Render for surface analysis
Appendix 10

The information in this section is taken verbatim from ArcGIS Desktop Help, under the Spatial Analyst, producing a hillshade.

Producing a hillshade

The Hillshade tool obtains the hypothetical illumination of a surface by determining illumination values for each cell in a raster. It does this by setting a position for a hypothetical light source and calculating the illumination values of each cell in relation to neighboring cells. It can greatly enhance the visualization of a surface for analysis or graphical display, especially when using transparency.

By default, shadow and light are shades of gray associated with integers from 0 to 255 (increasing from black to white).

The azimuth is the angular direction of the sun, measured from north in clockwise degrees from 0 to 360. An azimuth of 90 is east. The default is 315 (NW).

The altitude is the slope or angle of the illumination source above the horizon. The units are in degrees, from 0 (on the horizon) to 90 (overhead). The default is 45 degrees.
The hillshade below has an azimuth of 315 and an altitude of 45 degrees.

Using hillshade for display

By placing an elevation raster on top of a created hillshade and making the elevation raster transparent, you can create realistic images of the landscape. Add other layers, such as roads, streams, or vegetation, to further increase the informational content in the display.

Using hillshade in analysis

By modeling shade (the default option), you can calculate the local illumination and whether the cell falls in a shadow or not.
By modeling shadow, you can identify those cells that will be in the shadow of another cell at a particular time of day. Cells that are in the shadow of another cell are coded 0; all other cells are coded with integers from 1 to 255. You can reclassify all values greater than 1 to 1, producing a binary output raster. In the example below, the black areas are in shadow. The azimuth is the same in each image, but the sun angle (altitude) has been modified.