AN ABSTRACT OF THE THESIS OF

Joshua H. Petitmermet for the degree of Master of Science in Sustainable Forest Management presented on March 29, 2018.

Title: Tethering and Biochar: Two Emergent Technologies with Implications for Fuels Treatments on Federal Forest Lands

Abstract approved:

John Sessions

Two manuscripts are presented, each examining the potential impacts of emerging technologies on the economic feasibility of fuels reduction treatments on federal forest lands. Both manuscripts were prepared as part of the larger effort funded by the Institute for Working Forest Landscapes under the “Opportunities for biochar production to reduce forest wildfire hazard, sequester carbon, and increase agricultural productivity of dryland soils” project grant.

The first manuscript describes the cost and productivity of a tethered cut-to-length harvesting system operating on a steep slope fuels reduction treatment on the Fremont-Winema National Forest, establishing estimated costs per hour, per tonne, and per thousand board feet. Those costs are compared to estimates of cable logging systems. A variety of potential cycle time model forms are presented and evaluated.

The second manuscript explores the supply for and potential effects of biochar markets on the revenue generated from, and feasibility of, federal fuels reduction treatments. A suite of potential fuels reduction treatments is modeled for federally-owned forest land in the Upper Klamath basin using Forest Vegetation Simulator. The resulting outputs were used to optimize the modeled area for improvements to Composite Resistance Score under a variety of economic scenarios with a modified Great Deluge algorithm. Landscape level summary data is presented and discussed.

Tethering and Biochar: Two Emergent Technologies with Implications for Fuels Treatments on Federal Forest Lands

by

Joshua H. Petitmermet

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Master of Science

Presented March 29, 2018

Commencement June 2018

Master of Science thesis of Joshua H. Petitmermet presented on March 29, 2018.

APPROVED:

Major Professor, representing Sustainable Forest Management

Head of the Department of Forest Engineering, Resources, and Management

Dean of the Graduate School

I understand that that my thesis will become part of the permanent collection of Oregon State University libraries. My signature below authorizes release of my thesis to any reader upon request.

Joshua H. Petitmermet, Author

ACKNOWLEDGEMENTS

Given that the majority of those who made this work possible have already been included and recognized in the acknowledgements of the individual manuscripts, the author wishes to use this space to thank those who did not directly contribute, but whose presence and friendship maintained the author’s sanity over many long, hard years. In this vein, recognition goes to all past and present members of the World of Warcraft guild Lightbearer on Malygos-US, in particular Dan Anderson, Elizabeth Martin, Blythe and Trevor Robbins, and Ian Stroede. I couldn’t have done it without you.

Special recognition goes Gilder Turpskaday (also of Lightbearer), Lin-Manuel Miranda, and the cast of Hamilton. “Non-Stop” is the anthem to which this thesis was written.

Finally, thanks go also to Jennifer Petitmermet, Alicia and Colin Wilson and Cherie and Scot Yoder, only two of you are family by blood, but each of you is family in my heart.

CONTRIBUTION OF AUTHORS

Dr. John Sessions aided in refining methodology, writing, and editing of both chapter two and chapter three. Dr. John Bailey provided pre-treatment stand stocking data and assisted in the writing and editing of chapter two. Dr. Rene Zamora-Cristales assisted in the writing and editing of chapter two. Dr. Jeremy Fried provided technical support and aided in the writing and editing of chapter three.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

Page

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

LIST OF FIGURES

LIST OF FIGURES (Continued)

LIST OF FIGURES (Continued)

LIST OF TABLES

LIST OF TABLES (Continued)

LIST OF TABLES (Continued)

LIST OF APPENDIX FIGURES

LIST OF APPENDIX TABLES

TETHERING AND BIOCHAR: TWO EMERGENT TECHNOLOGIES WITH IMPLICATIONS FOR FUELS TREATMENTS ON FEDERAL FOREST LANDS

INTRODUCTION

Wildfire and the West

In the forests of western North America fire is a powerful and fundamental driver of ecological succession (Ryan et al. 2013). Fire regimes vary significantly from ecosystem to ecosystem, from the low frequency, high severity stand replacing events of the Oregon Coast Range (Long et al. 1998) to the low severity high or mixed frequency systems observed on the eastern slopes of the Cascades (Wright and Agee 2004). Historical causes have varied as well, driven in some regions by weather and lightning, driven in others primarily by human ignition as indigenous tribes seeking to improve acorn yields, hunting, and other ecosystem services (Pyne 2011), but in nearly every forest, fire has a significant role to play.

Settlement and Suppression

As European settlement displaced indigenous cultures, so too did it displace indigenous fire regimes. While fires were still sparked by the new arrivals via accident, fire-dependent aspen and oak systems began to degrade over time as the regular fires that maintain them largely ceased to occur (Pyne 2011). This process of degradation was reinforced by the development of federal fire suppression policies in the early twentieth century (Busenburg 2004), and exacerbated as aircraft became a common part of the wildland firefighting toolset in the 1940's, allowing fire suppression to reach even remote areas that would have been all but inaccessible before (Todd and Jewkes, 2006). Decades of highly effective fire suppression slowly led to significant increases in fuel loading and tree stocking, and corresponding vigor loss across large swaths of the forested North American west, turning much of it into a literal tinderbox.

Recent History

While the ecological importance of fire has been long understood, the reintroduction of fire has been a slow and fraught process. Early “let it burn” policies were blamed for the Yellowstone Fires of 1988, leading the National Park Service to halt all prescribed burn plans before the fires were even out (Ohmstead et al. 2012). Schindler and Toman (2003) found less than majority support for the use of prescribed fire in the Blue Mountains in 1996, and that that support dropped between 1996 and 2000.

But in the early 21st century that trend appears to have reversed. In 2004 Brunson and Schindler found greater than majority support for prescribed burning. In 2014 Toman et al. found that 85% of respondents indicated some level of support for prescribed fire. The rise of “megafires” may well be one of the culprits behind this change. Fires like the Biscuit (2002, approximately 500,000 acres burned), the Carlton Complex (2014, approximately 250,000 acres burned), and the Chetco-Bar Fire (2017, approximately 191,000 acres burned) have made headlines (and smoke) in the west summer after summer, and the public now believes prescribed fire to be an effective tool for reducing both fire risk and the cost of suppression (Toman et al. 2014).

Mechanical Fuel Reduction as an Alternative to Prescribed Fire

While it does not mimic the effects of fire for serotinous or other fire reliant species, mechanical fuels reduction treatments have consistently found higher levels of public support than prescribed fire (Schindler and Toman 2003, Brunson and Schindler 2004, Toman et al 2014). Mechanical treatments do not generate smoke like prescribed fire and have the potential to pay for themselves by selling the material collected. Further, Toman et al. (2014) found that respondents had more confidence in the ability of agency managers to use mechanical treatments effectively, and were more likely to see mechanical fuels reduction as an effective tool for reducing fire risk and suppression costs.

Mechanical treatments do have significant limitations though. Much of the west is mountainous, with large areas of forest occurring on steep slopes. For example, Jain et al. (2012) found that 41% of the forested ground in the Klamath Basin occurred on slopes of greater than

40%. Such slopes are generally considered too steep for traditional ground based treatment (North et al. 2015), and economically infeasible for treatment via cable harvest systems given the relatively high proportion of small diameter, low revenue material being removed. This is compounded by a general lack of data on the matter. Steep slope harvesting studies are rare, and those that do exist are often separated by hundreds or thousands of miles and were performed with different machines on different forest types with different silvicultural goals, making comparisons between them difficult (Lindroos and Cavalli 2016).

Opportunities for Improvement

Two emerging technologies appear to have the potential to significantly reshape the economics of steep slope mechanical fuels treatments: tethered cut-to-length systems and forest-origin biochar. Tethered cut-to-length systems have been observed by the author to operate on slopes of up to 90% with minimal difficulty, and existing literature (McIver et al. 2003, Flint 2013) suggests that cut-to-length systems could represent a dramatic reduction in cost when compared to cable systems operating under similar constraints.

Biochar has been the subject of several recent studies, assessing its properties as a soil amendment (Agnieszka 2016), climate change mitigation tool (Brassard et al. 2016), and as a soilless growth medium (Fornes and Belda 2017). Forest-origin biochar has the potential to improve or even create markets for small diameter wood that may otherwise be left on site or disposed (via pile and burn for example) of at a loss. It represents a potential for significant gains in both reducing the cost of mechanical fuels reduction treatments and increasing the revenue generated by it.

The IWFL Biochar Project

In May of 2015 a group of eight people representing Oregon State University (Drs. John Sessions, John Campbell, John Bailey, David Smith, and Steven Machado), the USDA Agricultural Research Service (Dr. Kristin Trippe), and the Oregon Department of Forestry (Marcus Kauffman) presented a proposal to the board of the Institute for Working Forest

Landscapes (IWFL). The group sought to study the potential for a “win-win-win” scenario in which:

- Tethered cut-to-length harvesting technology reduced the cost of federal fuels reduction treatments on steep slopes
- Biochar processing facilities provide a new market for the small diameter wood, increasing the revenues of those treatments, and
- Biochar made from those treatments could be applied in local agriculture to increase water holding capacity and crop productivity, and sequester carbon.

The Upper Klamath Basin of south-central Oregon and northern California was selected as a central area for the study, owing to the presence of large quantities of federally owned, dry, mixed-conifer forest suitable for fuels reduction treatments, and a significant quantity of agricultural land in which biochar could be applied.

Thesis Goals

The first goal of this thesis is to describe the observed productivity and cost of a steep slope fuels reduction treatment utilizing a tethered cut-to-length harvester/forwarder system. Data was collected over an observation period of three weeks in the summer of 2016 on a stand in the Fremont-Winema National Forest near Bly, Oregon. A variety of potential cycle time model forms are developed, evaluated, and discussed and a procedure for estimating productivity and cost are presented.

The second goal of this thesis is to study the effects of tethered cut-to-length costs and a set of potential biochar market scenarios on the feasibility of federal fuels reduction treatments in the Upper Klamath Basin of south-central Oregon and northern California, and to quantify the potential availability and cost of supply generated by those treatments. Treatments are applied, optimized, and analyzed for a variety of scenarios using a workflow that combines the Bioregional Inventory Originated Simulation Under Management (BioSum, Fried et al. 2017a) and a purpose-built heuristic optimization program (Neo-Processor).

Thesis Organization

This thesis is organized into four chapters including two manuscripts:

- Chapter 1 presents an introduction to the thesis organization, goals, scope, and context.
- Chapter 2 presents “Cost and Productivity of Tethered Cut-to-Length Systems in a Dry Forest Fuel Reduction Treatment, a Case Study”
- Chapter 3 presents “Assessment of the Potential Supply for Biochar Production and its Effect on the Feasibility of Federal Fuels Reduction Treatments in the Upper Klamath Basin” and
- Chapter 4 presents provides a synthesis of both manuscripts, provides conclusions, and outlines opportunities for future study.

COST AND PRODUCTIVITY OF TETHERED CUT-TO-LENGTH SYSTEMS IN A DRY FOREST FUEL REDUCTION TREATMENT, A CASE STUDY

Joshua H. Petitmermet, John Sessions, John Bailey, Rene Zamora-Cristales

Abstract

Fuels reduction treatments on steep slopes across federal forests of the western United States have been limited by the prohibitive costs associated with cable logging and poor market prospects for small diameter material. The emergence of tethered cut-to-length harvesting systems and small wood markets (e.g., biochar) could decrease costs and increase revenue generated. Over the course of three weeks we recorded a tethered cut-to-length fuels reduction treatment on the Fremont-Winema National Forest in south-central Oregon and used those data to derive hourly costs and productivity for a tethered harvester and forwarder. We also developed and tested a variety of cycle time model forms for each machine. Average utilization rate for the harvester was 87% on 28 untethered consolidated corridors and 64% on 17 tethered consolidated corridors. Similarly, the forwarder had an average utilization rate of 89% on 114 untethered bunks, and 76% on 30 tethered bunks. This reduced utilization rate could be due to the direct effects of tethering, the increased complexities of operations associated with steeper slopes, and the stand characteristics (e.g. lower density and tree sweep) of steeper slopes. Despite the loss of efficiency, our stump-to-truck costs do not exceed 50% of the pond value at the time of treatment, suggesting that the harvesting technology reported here was capable of paying for itself.

Introduction

Wildland fire is a subject of perennial concern in western North American forests. Most dry western forests evolved in concert with fire, adapting to it, or even becoming reliant on it to maintain some plant communities (Lotan 1976, Ryan et al. 2013). For most of the 20th Century, the dominant response to wildland fire was immediate suppression. Combined with grazing and other land management practices, particularly on federal lands, this led eventually to overstocking of trees, vigor loss, and ultimately increases in fire intensity, severity, and size. Though this change has been developing over the past few decades, there is a growing recognition of the detrimental effects of fire exclusion, opening the way for prescribed fire, wildland fire use, and fuels treatments to attempt to address the problem (Stephens and Ruth, 2005). However, in the mountainous west of the United States, many stands are considered unsuitable for fuels treatments due to the constraints imposed by steep slopes (North et al., 2015). The dry forests of the Klamath Basin in southern Oregon and northern California provide an example of this situation. Jain et al. (2012) rated 86.4% of forest land in the region as being in a “hazardous” state and found that 41% of the forest land in the area is on slopes of greater than 40%. In addition, no local small-diameter wood markets exist to absorb the lower quality material, requiring it to be piled and burned on site and further increasing the cost of treatment.

Two technologies are being studied in the hopes of finding solutions to these problems: tethered cut-to-length harvest systems and biochar production. Tethered cut-to-length systems may have the potential to bring a lower cost option to steep slopes, but few studies have been conducted on steep slope harvesting in general (Amishev et al. 2009, Ghaffariyan et al. 2012), much less tethered cut-to-length systems. Steep slope studies that have been conducted are widely spread around the world, making direct comparisons difficult (Lindroos and Cavalli 2016). Biochar, as a relatively new forest product used as a soil amendment in the horticultural and agricultural industries, could provide a market for small-diameter material that is otherwise currently non-merchantable. This study looks to fill in some of the gaps in both technologies, describing the cost, productivity, and behavior of tether-equipped cut-to-length systems on a fuel reduction treatment on the Fremont-Winema National Forest in south-central Oregon and, by extension, the cost and availability of potential biochar feedstock under those same conditions. Cut-to-length operations using harvesters and forwarders are a small but growing harvest method in Oregon. This study documents the first tethered harvester-forwarder operation on the Fremont-Winema National Forest, and this particular sale was modified to permit this experiment.

Methods

We gathered data in two distinct ways, direct observation of harvest operations and direct correspondence with both the logging contractor observed (Miller Timber Services, Philomath, Oregon) and a representative of the equipment manufacturer (Ponsse North America, Coburg, Oregon).

Direct Observation

Field observations were recorded between 12 July, 2016 and 29 July, 2016 on Pilot Project Unit 10 on the Bly Ranger District in south-central Oregon. These observations encompass all felling, processing, decking, yarding, and piling activities on 10.9 hectares (27 acres), approximately 43% of the 25.5 hectares (63 acres) that were treated in total. The unit is a dry mixed-conifer stand ranging between 1800 and 1900 meters in elevation. Slopes ranged from 12 to 70 percent with an average slope of 38 percent. Soils were primarily loamy-skeletal, derived from a parent material of ash over top of basalt (R. Rone, personal communication, 2017). No precipitation occurred during the period of observation. The pre-treatment stand was dominated by white fir (*Abies concolor*) and ponderosa pine (*Pinus ponderosa*) with scattered conifers of other species.

The pre-treatment stand averaged 33.3 square meters of basal area per hectare (145 square feet of basal area per acre), with a treatment goal of thinning down to an average of 11.5 square meters of basal area per hectare (50 square feet per acre) with a “clumpy” distribution of leave trees throughout the landscape (Figure 2.1). Trees were preferentially chosen for removal by size, species, and vigor. Smaller trees were preferentially removed, to a maximum take diameter of 53.34 centimeters (21 inches). Species removal preferences targeted western juniper (*Juniperus occidentalis*), lodgepole pine (*Pinus contorta*), white fir, incense-cedar (*Calocedrus*

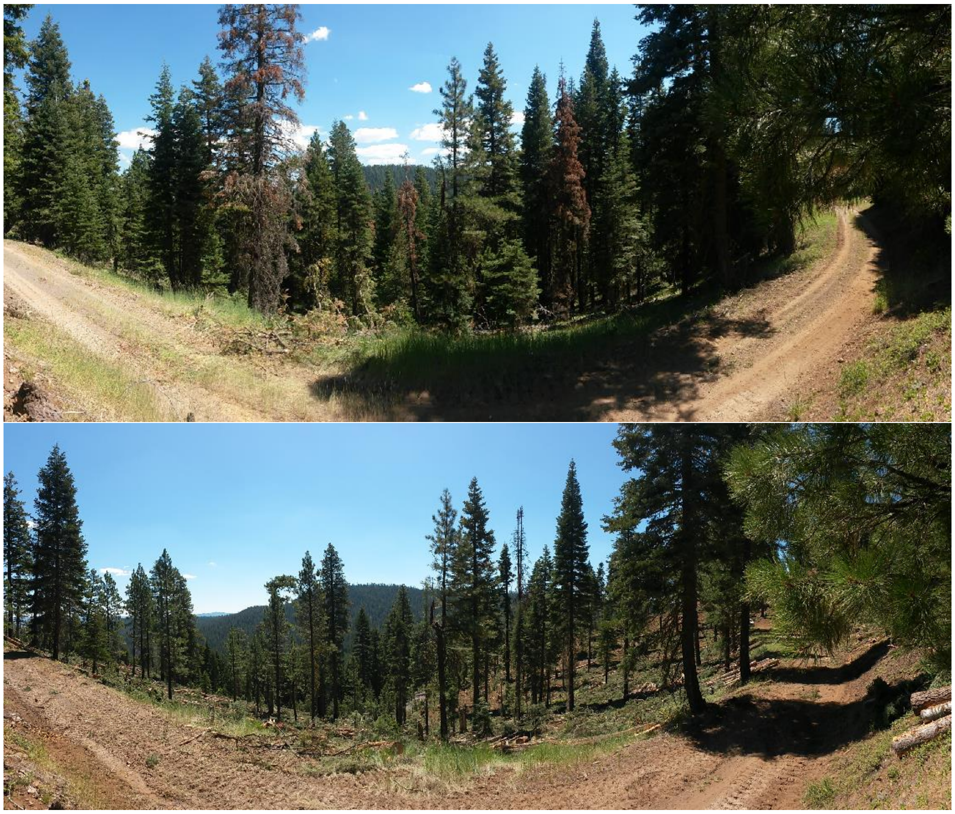


Figure 2.1: Stand conditions before treatment (above) and after treatment (below) on Pilot Project Unit 10 on the Fremont-Winema National Forest

decurrens), and ponderosa pine, in that order. Unhealthy trees were also preferentially removed, except for five needle pines (*Pinus lambertiana* and *Pinus monticola*), which were retained regardless of all other considerations.

Cutting and processing was done with a Ponsse Bear harvester (Figure 2.2). The Ponsse Bear weighs approximately 24.5 tonnes with 240 kW of engine power and a C6 boom with a reach of 10 meters. Logs were extracted with a Ponsse Elephant King forwarder. The Ponsse Elephant King has 210 kW of engine power with a boom reach of 9.5 meters and is rated for a maximum carrying capacity of 20 tonnes. Both machines use an eight-wheeled double-bogey design. Each machine was run by a single, highly experienced operator for the duration of observation. Both machines were tether equipped; when and where each machine would tether was left to each operator's discretion. A few trees were marked on the first day of operations to aid the harvester operator in identifying which trees to cut and the desired density of tree retention. After the first day neither corridors nor individual trees were pre-selected or marked, leaving decisions regarding navigation and tree selection to the harvester operator's discretion. In general, the forwarder would follow in the harvester's path, but only after the harvester had moved onto another corridor. All forwarding for the duration of observation was adverse (uphill travel when loaded) or done along roads of low gradient. The majority of logs were taken to one of ten roadside decks to await truck loading. All log transport was done by straight bed trucks with log bunks pulling a short log trailer (maximum log length of roughly 8 meters) with a maximum log capacity of approximately 30 tonnes. All loading was performed by the forwarder.

Commented [JS1]: A photo of the operation of the harvester and forwarder would help readers, particularly if they are tethered.

Photos of the before and after would also be informative either in this manuscript, the next or even in the Chapter 4 or 5.

Operations were recorded using a pair of hat-mounted GoPro Hero Silver 4 cameras, one worn by each operator for the duration of each shift, excluding fire watch (a period of one to two hours at the end of each shift spent watching for signs of any accidental ignitions that could have occurred during that shift). Each camera used on-board power via a USB cable to negate the need to change batteries. Recordings were saved in-camera to Samsung Pro Plus 128GB MicroSDXC memory cards. Memory cards were changed twice per day, once at midday and once at shift's end. Each time memory cards were changed, the video on the card was transferred to one of several external hard drives. The camera for each machine was turned on when that machine was turned on for the day, and turned off when that machine was shut down at the end of each shift.

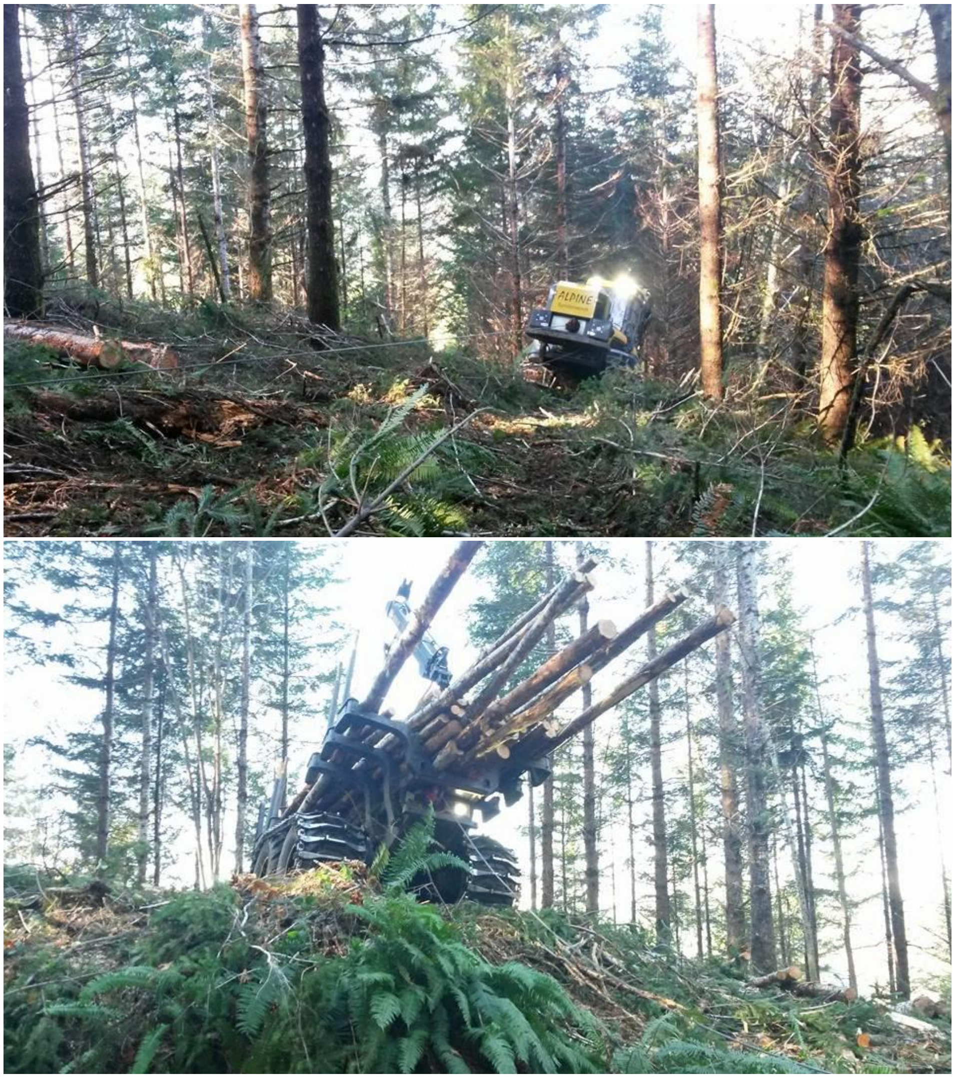
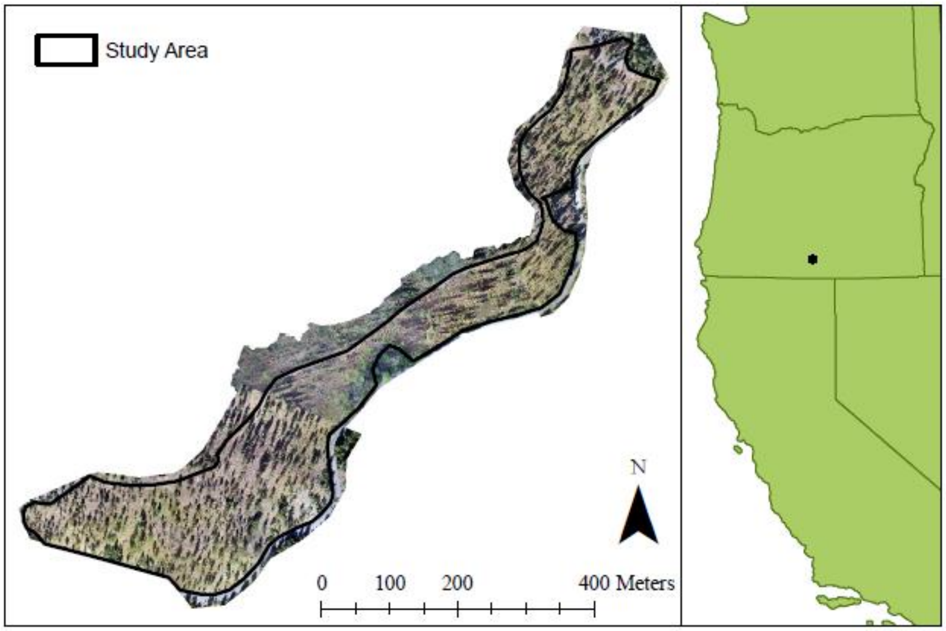


Figure 2.2: Ponsse Bear harvester (above) and Elephant King forwarder (below) on tethered thinning operations in the Oregon Coast Range

To aid in corridor identification and measurement, the treatment area was flown over via drone on 8 August, 2016, and the resulting images were stitched together into a georeferenced photograph (Figure 2.3).



State boundary data taken from United States Geological Survey
National Boundary Dataset, available online at [viewer.nationalmap.gov]

Figure 2.3: Study location and area of observed operations

Correspondence

To obtain information that could not be directly observed in the field, the authors corresponded with both a representative of the manufacturer and a representative of the logging contractor. In general, the manufacturer was asked for specific information pertaining to machine costs and the contractor was invited to comment on that information. Much of the requested data is highly variable and/or considered confidential, so the numbers presented here should be treated as reasonable approximations, not exact values. To describe some of this uncertainty, the information from correspondence was used to generate a “low cost” machine rate scenario and “high cost” machine rate scenario using a modified version of the machine rate method described by Brinker et al. (2002).

Field Data Processing

Recordings were processed manually using Windows Media Player. Each video was broken down into three components: active time, delay time, and excise time. Active time was defined as all time in which some part of the machine was in motion, including any delays of less than thirty seconds. Delay time was defined as all time in which the machine was inactive for longer than thirty seconds, including breaks, lunches, and both administrative and mechanical delays. Excise time was defined as all delays of any length that were incurred as a direct result of this study. In general, excise time consisted of five to six minutes per machine per day, the time required to change memory cards in each camera. Excise time was excluded from analysis.

The time for each video was further broken down into units of interest. For the harvester, the sole unit of interest was the corridor. For the forwarder, the units of interest were the bunk (while forwarding) and the truck (while loading short log trailers). Forwarder bunks were further identified as being one of three types: productive, deck (pile) consolidation, and carriage. Productive bunks are all those where previously ungathered material is collected and deposited to a deck or roadside. Deck consolidation bunks are all those where the forwarder moves the remnants of a pre-existing roadside deck to another deck, generally when less than a single truck load remains in the first deck. Carriage bunks are a specific case of deck consolidation bunk created when the forwarder loads several bunks on short, steep corridors without untethering, unloading those bunks temporarily at roadside and then returning to reload and move that material to a deck. Time spent on deck consolidation bunks was prorated to all productive bunks to account for that time in a uniform manner; time spent on carriage bunks was prorated among all productive bunks in the corridor that produced those carriage bunks.

A tally of pieces loaded and the associated corridor of origin was kept for each bunk and truck. Each piece counted was classified as a saw log, or a “feedstock” log. Saw logs were defined as those of commercial species and grades acceptable to local mills, predominantly grand fir and ponderosa pine with minimal defect to a 15.24-centimeter (six-inch) top. Feedstock logs are all those of species or conditions unacceptable to local mills that could be transported in log form for use as biochar feedstock. This includes logs of any species to a 10.16-centimeter (four-inch) top, and saw logs rejected due to sweep, damage, or defect. Due to the lack of an existing pulp or biochar market in the region, feedstock logs were loaded onto trucks for weighing, and then unloaded into piles for burning at a later date.

Only the front bunk of each short log trailer was used for feedstock weighing, and time spent unloading feedstock logs after weighing was treated as excise time for the purposes of analysis. Utilization rates, defined as the active time divided by the total time, were calculated for each corridor and bunk observed. An average utilization rate for each machine was calculated as the time-weighted average of those observed utilization rates. Utilization rates were not calculated for truck loading. The majority of observed delay time associated with truck loading was administrative; time used by the operator filling out forms for the mill and landowner. Since all feedstock logs remained on site, those delays only occurred on saw log loads, creating the illusion of a significant gap in utilization rates. To address this, all truck loading rates were calculated in terms of productive hours only.

The piece counts and truck weights were used to calculate an average piece weight for each material type by pile and for the stand as a whole. The stand average values were then used to calculate the weight of material removed by the forwarder with each bunk (aka “turn” or “trip”) as well as the weight of material produced by the harvester in each corridor. In several cases corridors had to be consolidated for analysis due to difficulty in determining the weight associated with that corridor. Corridors were only consolidated if they were adjacent to each other and either all tethered or all untethered. Individual bunk information was also summed for each corridor for the purpose of cycle time analysis. The bunk may be the most intuitive modeling unit for the forwarder, but multiple bunks in the same corridor violate any assumption of independence among observations. What the forwarder does on one bunk in a corridor inherently determines the distance traveled and the quantity and type of material collected for all subsequent bunks in that same corridor.

All cubic volume estimates were calculated as a function of green weights using a conversion factor. The conversion factor was calculated using equations based on those used by the USDA Forest Service FIA (Forest Inventory and Analysis) methodology for calculating species specific bark and wood weights from green volumes. Specifically:

(1) $wvol = tweight / (1 + mc) / (wspgr * con) * (1 – (bvp / (1 + bvp)))$

(2) $bvol = tweight / (1 + mc) / (bspgr * con) * (bvp / (1 + bvp))$

Where: bvol and wvol are the volume of bark and wood respectively,

tweight is the total weight of material being converted

mc is the dry-basis moisture content expressed as a fractional value

bvp is the volume of bark as a percent of the wood volume

wspgr and bspgr are the specific gravities for wood and bark of a given species respectively, and

con is the reference constant associated with those specific gravities.

All specific gravities, bark volume ratios, and the reference constant were taken directly from the values provided in the most recently released FIA dataset for the state of Oregon (United States Department of Agriculture, 2017). An assumed average moisture content of 72% dry basis (42% wet basis) was used for all species. With an estimated species mix of 70% white fir and 30% ponderosa pine, we obtained a conversion ratio of 1.53 cubic meters per green tonne (48.94 cubic feet per green US ton).

All board foot volume estimates were calculated as a function of cubic volume estimates, using the board feet per cubic foot of bolewood inside bark ratios established by Keegan et al. (2010). Specifically, we used the Westside Scribner ratio for the state of Oregon: 148 board feet per cubic meter (4.19 board feet per cubic foot). Applied to the wood-only volume estimates of our 70/30 species mix this results in 0.199 thousand board feet (MBF) per green tonne (0.181 MBF per green US ton). This may be a significant underestimation of the locally scaled volume. Both Eastside and Westside Scribner rules are used in Oregon, with the crest of the Cascade mountain range as the line of demarcation between them (Fonesca 2005). Both our study site and the mill to which the saw material was sent were east of the Cascades, but there is less peer-reviewed Oregon-specific Eastside data to draw from. If we look again to Keegan et al. (2010), the closest substitute geographically would be the Eastside Scribner ratio for California: 177 board feet per cubic meter (5.02 board feet per cubic foot). Applied to the wood-only volume estimates of our 70/30 species mix the California ratio would result in 0.238 MBF per green tonne (0.216 MBF per green US ton), a twenty percent increase in estimated scaled volume which would result in a 16% decrease in our estimated cost per MBF when compared to the Westside Oregon ratio. When comparing costs to other harvesting studies that use board foot scale, addressing log scale in comparable units is important. In this study we are reporting Westside scale but also later include both Westside and Eastside scale when drawing comparisons to local log values.

The length of each corridor and transit distance for each bunk were measured in ArcMap 10.4 using post-treatment drone photos and, where necessary, landmarks identified in the harvester and forwarder videos. The length of a corridor was defined as the total distance traveled between the start of one corridor and the next, with the start of each corridor being defined by the first tree cut or the first harvester tread going off road, whichever occurred first. The transit distance for each bunk was defined as the total pile-to-pile distance, starting from the pile where the last log of the previous bunk was unloaded to the pile where the last log of the current bunk was unloaded.

After processing, the information for both machines was imported into RStudio for cycle time modeling. These cycle time models are intended to find an effective means of estimating the time required to treat a given area and to answer a key question: does tethering behave like a fixed or variable cost? Can it be accurately described solely by the delay time required to tether and untether, or is there an ongoing productivity cost incurred by reduced mobility and/or increased difficulty in material handling while tethered?

The following model forms were considered for both machines:

(3) $PMM = \beta_1 \, ctot + \beta_4 \, cdis$

(4) $PMM = \beta_2 \, utot + \beta_3 \, ttot + \beta_4 \, cdis$

(5) $PMM = \beta_1 \, ctot + \beta_5 \, udis + \beta_6 \, tdis$

(6) $PMM = \beta_2\, utot + \beta_3\, ttot + \beta_5\, udis + \beta_6\, tdis$

(7) $PMM = \beta_4\, cdis + \beta_7\, csaw + \beta_{10}\, cbf$

(8) $PMM = \beta_5\, udis + \beta_6\, tdis + \beta_7\, csaw + \beta_{10}\, cbf$

(9) $PMM = \beta_4\, cdis + \beta_8\, usaw + \beta_9\, tsaw + \beta_{11}\, ubf + \beta_{12}\, tbf$

(10) $PMM = \beta_5\, udis + \beta_6\, tdis + \beta_8\, usaw + \beta_9\, tsaw + \beta_{11}\, ubf + \beta_{12}\, tbf$

Where: PMM is the time required to complete the corridor, in productive minutes

ctot is the total weight of material produced in the corridor, in green tonnes

cdis is the total distance traversed in the corridor, in meters

utot is the weight of material produced while untethered, in green tonnes

ttot is the weight of material produced while in tethered, in green tonnes

udis is the untethered distance traversed in the corridor, in meters

tdis is the tethered distance traversed in the corridor, in meters

csaw is the weight of saw-log material produced in the corridor, in green tonnes

cbf is the weight of biochar feedstock produced in the corridor, in green tonnes

usaw is the weight of saw-log material produced while untethered, in green tonnes

tsaw is the weight of saw-log material produced while tethered, in green tonnes

ubf is the weight of biochar feedstock produced while untethered, in green tonnes

tbf is the weight of biochar feedstock produced while tethered, in green tonnes

Models were named H3 through H10 for the harvester and F3 through F10 for the forwarder, corresponding to the equation numbers listed above. The base harvester and forwarder models were forced through the origin under the assumption that if no distance is traveled and no material is gathered, no work is actually being done and the machine time required is zero. A series of functionally identical models with an intercept, denoted H3I, F3I, and so on, will also be presented to assess likelihood that this assumption is correct for our data. All model comparisons will be made using second-order Akaike's Information Criterion (AIC_C). AIC_C predicts the relative likelihood of one model being correct for a given data set when compared to another, and is expressed as both an AIC_C score and an evidence ratio (Motulsky and Christopoulos, 2004).

While the exclusion of an intercept makes logical sense, it does present analytical challenges including inflated adjusted R^2 values and F-statistics when compared to similar models with an intercept. How to properly assess the fit of such models has been and continues to be debated (Eisenhauer 2003), and is beyond the scope of this study. As a result, the strength of fit for the no-intercept models presented here should only be considered relative to each other.

All model fitting was performed with RStudio's built in lm() function. AIC_C scores were computed using the AIC_C() function from the MuMIn R package. All p-values reported in relation to individual variables were generated via t-test by the lm() function and are equivalent to the p-value obtained by an extra sum of squares F-test comparing models with and without that variable.

Machine Costing

Machine costs were calculated using the machine rate method described by Brinker et al. (2002) with two differences. First, where the machine rate method calculates fuel consumption as a function of engine horsepower, we used the average rate of fuel consumption observed during field data collection. Second, we added an additional calculation block to account for costs associated with fire watch. Fire watch costs were calculated as:

(11) ft = SMH * ft%

(12) fh = (ft / asl) * fl

(13) ff = fh / SMH

(14) fcSMH = ff * WB

(15) fcPMH = fcSMH / ut%

Where: SMH is the scheduled machine hours per year ft% is the percent of yearly scheduled hours subject to fire watch requirements

ft is the scheduled machine hours subject to fire watch requirements

asl is the average shift length in hours

fl is the average length of fire watch in hours

fh is the number of operator hours per year spent on fire watch

ff is the hours of fire watch per scheduled machine hour

WB is the operator wage and benefit cost in dollars per scheduled machine hour

fcSMH is the cost of fire watch in dollars per scheduled machine hour

ut% is the utilization rate, and

fcPMH is the fire watch cost in dollars per productive machine hour.

This method effectively prorates the cost of fire watch across all scheduled machine hours in a given year, allowing the fire watch cost to be added to the existing cost per scheduled machine hour / cost per productive machine hour framework without additional modification. Machines are required to be shut down for the duration of fire watch, so fire watch costs are calculated solely as a function of labor costs, and assume no additional depreciation or other losses of value during that time.

This method may slightly overestimate the cost of fire watch because it cannot account for opportunistic work done during fire watch. Operators often used a portion of fire watch time to clean, perform basic maintenance, and/or scout the ground for the next day's work, all activities that would otherwise require time that the machine could have been in operation.

As discussed previously, we produced two sets of input for cost estimation (Table 2.1) to better reflect the uncertainty in and variability of inputs taken from correspondence with the equipment manufacturer and logging contractor.

Table 2.1: Machine rate method inputs by machine and machine rate scenario

Machine	Harvester		Forwarder	
Machine Rate Scenario	High Cost	Low Cost	High Cost	Low Cost
Purchase Price ($)	950,000	850,000	750,000	650,000
Machine Life (Years)	5	5	5	5
Salvage Rate (%)	40	50	40	50
Utilization Rate (%)	78	78	86	86
Repair and Maintenance (%)	30	15	30	15
Interest (% of Avg Investment)	10	10	10	10
Insurance/Tax (% of Avg Investment)	4	4	4	4
Fuel Consumption (Liters per Hour)	40.54	40.54	27.90	27.90
Fuel Cost ($ per Liter)	0.64	0.64	0.64	0.64
Lube and Oil (%)	36.7	36.7	36.7	36.7
Wage/Benefit ($/Hour)	35	35	35	35
Scheduled Hours (Hours per Year)	2000	2000	2000	2000
Yearly Fire Watch (%)	30	30	30	30
Average Shift Length (Hours)	12	12	12	12
Average Fire Watch Length (Hours)	1.5	1.5	1.5	1.5

Results and Discussion

Utilization and Productivity

A total of 107 hours of video across 45 consolidated corridors (made from 61 individual corridors) was collected and analyzed for harvester operations. The harvester demonstrated an average utilization rate of 87% on the 28 fully untethered consolidated corridors and 64% on the 17 fully or partially tethered consolidated corridors, providing an overall average utilization rate of 78%. The tethered utilization may be an underestimation of the true average due to three long duration delays that occurred during tethered observation; a cable break, a fire weather shut down, and a hose break that required driving to the nearby town of Lakeview, Oregon for a replacement part. Those three incidents alone accounted for a nearly six and a half hours of delay, more than a quarter of the total delay time observed for the harvester.

For the forwarder, a total of 131 hours of video was collected and analyzed, comprising 95 hours of forwarding across 144 bunks and 36 hours of loading across 49 trucks. Of 144 bunks, 131 were productive bunks, with eight pile consolidation bunks and five carriage bunks.

The forwarder demonstrated an average utilization rate of 89% on the 114 untethered bunks, 76% on the 30 tethered bunks, yielding an average of 86% overall. On average, each bunk carried 11.9 tonnes, with an average of 11.1 tonnes per bunk while tethered and 12.2 tonnes per bunk while untethered. In nearly all cases the limiting factor appeared to be bunk volume, rather than load weight, regardless of tether status. Of 49 truck loadings observed, 36 were saw log loads and 13 were feedstock log loads. According to our per-bunk and per-corridor calculations, the total weight of feedstock logs removed in the area of observation was equal to 27% of the total weight for all material. According to the final weight tallies and mill receipts for all 25.5 ha (63 acres) treated, the total weight of feedstock logs (812 green tonnes) produced during treatment was 25% of the total for all material (3,225 green tonnes).

Both the harvester and forwarder demonstrated a notable drop in average productivity while tethered (Table 2.2), but it is important to note that this may not be a direct result of tethering itself. The steeper sections of the site often had lower stocking, smaller trees, and intermittent rock outcroppings, each of which being likely to reduce productivity independently of slope or tether use.

Table 2.2: Observed rates of machine utilization and production

		Tethered	Untethered	Average
Harvester Utilization	% Active Time	64	87	78
Harvester Production	Green tonnes per SMH	11.4	17.0	14.7
	m^3 per SMH	14.5	25.2	20.8
	MBF per SMH	1.9	3.3	2.7
Forwarder Utilization	% Active Time	76	89	86
Forwarder Production	Green tonnes per SMH	13.5	17.8	16.7
	m^3 per SMH	20.6	27.2	25.5
	MBF per SMH	2.7	3.6	3.3

		Feedstock Logs	Saw Logs	Average
Forwarder Loading	Green tonnes per PMH	35.3	59.4	48.4
	m^3 per PMH	53.9	90.7	73.9
	MBF per PMH	7.0	11.8	9.7

Lower productivity is also seen when comparing the loading of feedstock and saw logs on to trucks, but the explanation in that case is much clearer. Feedstock logs are considerably smaller and more irregular in size and shape, requiring more time and care in grabbing, moving, and packing. Similarly, feedstock logs are far more likely to be "fumbled" and drop out of the forwarder's grip while in motion, requiring a second motion to retrieve it.

Machine and Material Costs

The high cost machine rate scenario generated harvester costs of $188 and $242 per scheduled and productive machine hour respectively and forwarder costs of $160 and $205 per scheduled and productive machine hour respectively. The low cost machine rate scenario generated harvester costs of $155 and $181 per scheduled machine hour and productive machine hour respectively and forwarder costs of $131 and $152 per scheduled and productive machine hour respectively. These scheduled and productive hour costs do not include contractor profit and risk allowance, supervision and administration, or the fixed cost of equipment mobilization. These costs do include our firewatch cost estimate of $1.31 per scheduled hour for both machines ($1.68 and $1.53 per productive machine hour for the harvester and forwarder respectively).

If we assume that both feedstock and saw logs can be sold, the stump-to-truck costs (including loading) are $27.30 and $25.22 per green tonne ($24.77 and $22.88 per green US ton) for feedstock and saw logs respectively under the high cost machine rate scenario and $23.09 and $21.34 per green tonne ($20.95 and $19.36 per green US ton) for feedstock and saw logs respectively under the low cost machine rate scenario. On a cubic volume basis, this results in estimated stump-to-truck costs of $17.87 and $16.51 per cubic meter ($0.51 and $0.47 per cubic foot) for feedstock and saw material respectively under the high cost machine rate scenario, and $15.12 and $13.97 per cubic meter ($0.43 and $0.40 per cubic foot) under the low cost machine rate scenario. On a board foot volume basis, this results in estimated stump-to-truck costs of $137 and $126 per MBF of feedstock and saw logs respectively under the high cost machine rate scenario and $116 and $107 per MBF of feedstock and saw logs respectively under the low cost machine rate scenario. We recognize that the use of an average board-foot scaling conversion underestimates the cost differences between the feedstock and saw logs on a board-foot basis as the saw log is larger than the feedstock material.

When there is no market for feedstock logs, as there was not at the time of observation, that feedstock material is piled and burned at roadside, incurring an additional burning cost of approximately $13.34 per hectare (P. Cheng, personal communication, 2017). If we assume that the saw logs must also bear the costs of cutting, gathering, piling, and burning the feedstock material (business as usual), the estimated stump-to-truck cost of saw log material rises to $32.75 per tonne, $21.44 per cubic meter, and $164 per MBF under the high cost machine rate scenario and $27.74 per tonne, $18.16 per cubic meter, and $139 per MBF under the low cost machine rate scenario. The estimation of these “business as usual” costs was not the original intent of the study, but they are a useful benchmark for comparison. The business as usual cost serves as the most accurate estimate of stump-to-truck treatment costs at the time of the study and as a worst case (highest cost) scenario for the immediate future, should a market for feedstock material fail to develop.

Material Cost Assessment

Making effective comparisons from one cost or productivity study to another is always difficult. Not only do forests, prescriptions, and harvest systems vary from region to region and over time, but published studies are few. Cavalli’s 2011 literature review found only three empirical cable yarding studies in the United States over an eleven year period (Lindroos and Cavalli, 2016). Studies on thinning productivity with cut-to-length systems are also uncommon. Flint (2013) studied the difference between cable yarding and untethered cut-to-length operations in the Oregon Coast Range but, due to methodology used, provided only estimated costs on a dollars per truck load basis. Prior to that, McIver et al. (2003) compared the effectiveness of a harvester/forwarder and harvester/cable systems on flat ground (average slope < 13%) in northeast Oregon, and Kellogg and Bettinger (1994) studied low-slope cut-to-length thinning in the western foothills of the Cascades near Lyons, Oregon. Neither is ideal for direct comparison, but there are other measures to evaluate against, including regional surveyed logging cost estimates and the local pond values for the material being gathered.

In 2014, sixteen “knowledgeable and reputable” logging operators were surveyed as part of an effort to update the stump-to-truck cost estimates used by the Oregon Department of Forestry (ODF) for their strategic planning efforts in western Oregon (Christian, 2014). The responses from that survey were used to construct harvest cost estimates for regeneration and partial cut harvests under a variety of indicators and harvest difficulty classifications. These classifications include yarding distance, volume removed per hectare, percent slope, and average diameter at breast height. Using that harvesting cost estimating system with the characteristics of our study site provides an estimated ground-based harvest cost of $205 per MBF (Westside Scribner) 24% higher than our business-as-usual high cost scenario price per MBF and 47% higher than our business-as-usual low cost scenario price per MBF (Westside Scribner). This difference is, at least in part, due to cost accounting differences. Christian (2014) included facility maintenance, administrative expenses, and a 10% risk factor in his stump-to-truck costs and had 12% higher diesel fuel rates. Using Christian’s assumptions to predict the cost of cable harvest under the same conditions results in an estimated price of $334 per MBF, 75% higher than the estimated cost of ground-based harvest and 103% and 140% higher than the prices calculated under the business-as-usual high and low cost scenario respectively. Flint (2013) found cost differences of a similar magnitude between conventional manual felling and cable logging compared to use of harvester-forwarders, with the cost per load of his manual felling with cable yarding treatment 98% higher than the cost per load of his cut-to-length harvester-forwarder treatment with an adverse haul. McIver et al. (2003), despite using a harvester for felling, also found a similar relationship on flat ground, with cable units averaging costs per ton 74% higher than forwarder units.

According to recently published data (Oregon Department of Forestry, 2017), pond values for the Klamath region at the time of our study ranged from $335 to $375 per MBF (Eastside Scribner) for ponderosa pine and $395 to $415 per MBF (Eastside Scribner) for true fir, with prices increasing with increases in top end log diameter inside bark. With our 70/30 species mix of white fir and ponderosa pine that would equate to average pond value ranging from $377 to $390 per MBF (Eastside Scribner). Our stump-to-truck costs do not exceed 50% of the pond value at the time of treatment when using either the Eastside California conversion ratio (Table 2.3) or the Westside Oregon ratio (Table 2.4). These stump-to-truck costs do not represent the full cost of treatment, but may indicate that the treatment was capable of paying for itself when using the harvesting technology reported here. For that to not be the case, the sum of uncalculated costs (including mobilization, truck transport, road maintenance, administration, contractor profit and risk, and stumpage) would need to be more than the stump-to-truck costs under all machine rate scenarios, and double or triple the stump-to-truck costs when using the Eastside conversion ratio and/or the low cost machine rate scenario.

Table 2.3: Estimated stump-to-truck costs as a percent of saw log pond values (W)

Average top end log diameter inside bark		cm	15.24 - 20.32	20.32 - 35.56	35.56 - 55.88
Pond value (pv)		$ per MBF[a]	377	382	390
Low cost machine rate scenario	Business-as-usual[b]	144	38%	38%	37%
	Feedstock logs	120	32%	31%	31%
	Saw logs	111	29%	29%	28%
High cost machine rate scenario	Business-as-usual[b]	171	45%	45%	44%
	Feedstock logs	142	38%	37%	36%
	Saw logs	131	35%	34%	34%

[a] Pond values are in $/MBF Eastside Scribner while cost estimates are in $/MBF Westside Scribner

[b] Business-as-usual assumes that saw logs must carry the cost of feedstock log recovery and disposal

Table 2.4: Estimated stump-to-truck costs as a percent of saw log pond values (E)

Average top end log diameter inside bark		cm	15.24 - 20.32	20.32 - 35.56	35.56 - 55.88
Pond value (pv)		$ per MBF[a]	377	382	390
Low cost machine rate scenario	Business-as-usual[b]	121	32%	32%	31%
	Feedstock logs	100	27%	26%	26%
	Saw logs	93	25%	24%	24%
High cost machine rate scenario	Business-as-usual[b]	143	38%	37%	37%
	Feedstock logs	118	31%	31%	30%
	Saw logs	109	29%	29%	28%

[a] Pond values and cost estimates are both in Eastside Scribner

[b] Business-as-usual assumes that saw logs must carry the cost of feedstock log recovery and disposal

Cycle Time Model Results and Implications

The cycle time modeling produced a dichotomy between machines. The harvester model was strongest with the simplest model (Table 2.5 and Table 2.6). Using only total weight of

Table 2.5: Summary of base harvester models

Model	Adj-R^2	Variable	ctot	utot	ttot	cdis	udis	tdis	csaw	usaw	tsaw	cbf	ubf	tbf
		β	1	2	3	4	5	6	7	8	9	10	11	12
		Units	gt	gt	gt	m	m	m	gt	gt	gt	gt	gt	gt
H3	0.9715	Coefficient	1.43	---	---	0.25	---	---	---	---	---	---	---	---
		P-Value	<0.001	---	---	<0.001	---	---	---	---	---	---	---	---
H4	0.971	Coefficient	---	1.45	1.36	0.25	---	---	---	---	---	---	---	---
		P-Value	---	<0.001	<0.001	<0.001	---	---	---	---	---	---	---	---
H5	0.971	Coefficient	1.41	---	---	---	0.25	0.24	---	---	---	---	---	---
		P-Value	<0.001	---	---	---	<0.001	<0.001	---	---	---	---	---	---
H6	0.9703	Coefficient	---	1.43	1.38	---	0.25	0.25	---	---	---	---	---	---
		P-Value	---	<0.001	<0.001	---	<0.001	<0.001	---	---	---	---	---	---
H7	0.9708	Coefficient	---	---	---	0.25	---	---	1.43	---	---	1.4	---	---
		P-Value	---	---	---	<0.001	---	---	<0.001	---	---	0.017	---	---
H8	0.9703	Coefficient	---	---	---	---	0.25	0.24	1.42	---	---	1.36	---	---
		P-Value	---	---	---	---	<0.001	<0.001	<0.001	---	---	0.025	---	---
H9	0.9695	Coefficient	---	---	---	0.25	---	---	---	1.46	1.37	---	1.37	1.27
		P-Value	---	---	---	<0.001	---	---	---	<0.001	<0.001	---	0.031	0.275
H10	0.9687	Coefficient	---	---	---	---	0.25	0.25	---	1.45	1.38	---	1.36	1.32
		P-Value	---	---	---	---	<0.001	<0.001	---	<0.001	<0.001	---	<0.046	0.342

Table 2.6: Summary of harvester models with intercept

Model	Adj-R^2	Variable	Intercept	ctot	utot	ttot	cdis	udis	tdis	csaw	usaw	tsaw	cbf
		β	0	1	2	3	4	5	6	7	8	9	10
		Units	NA	gt	gt	gt	m	m	m	gt	gt	gt	gt
H3I	0.9412	Coefficient	8.11	1.38	---	---	0.24	---	---	---	---	---	---
		P-Value	0.138	<0.001	---	---	<0.001	---	---	---	---	---	---
H4I	0.9412	Coefficient	10.03	---	1.41	1.21	0.23	---	---	---	---	---	---
		P-Value	0.085	---	<0.001	<0.001	<0.001	---	---	---	---	---	---
H5I	0.9412	Coefficient	9.95	1.33	---	---	---	0.25	0.22	---	---	---	---
		P-Value	0.087	<0.001	---	---	---	<0.001	<0.001	---	---	---	---
H6I	0.9399	Coefficient	10.14	---	1.38	1.25	---	0.24	0.23	---	---	---	---
		P-Value	0.087	---	<0.001	<0.001	---	<0.001	<0.001	---	---	---	---
H7I	0.9398	Coefficient	8.12	---	---	---	0.24	---	---	1.38	---	---	1.34
		P-Value	0.143	---	---	---	<0.001	---	---	<0.001	---	---	0.02
H8I	0.9398	Coefficient	10.02	---	---	---	---	0.25	0.22	1.35	---	---	1.22
		P-Value	0.09	---	---	---	---	<0.001	<0.001	<0.001	---	---	0.039
H9I	0.9384	Coefficient	10.17	---	---	---	0.24	---	---	---	1.43	1.24	---
		P-Value	0.09	---	---	---	<0.001	---	---	---	<0.001	<0.001	---
H10I	0.9368	Coefficient	10.25	---	---	---	---	0.24	0.23	---	1.41	1.26	---
		P-Value	0.09	---	---	---	---	<0.001	<0.001	---	<0.001	0.001	---

Table 2.6: Summary of harvester models with intercept (Continued)

Model	Adj-R^2	Variable	ubf	tbf
		β	11	12
		Units	gt	gt
H3I	0.9412	Coefficient	---	---
		P-Value	---	---
H4I	0.9412	Coefficient	---	---
		P-Value	---	---
H5I	0.9412	Coefficient	---	---
		P-Value	---	---
H6I	0.9399	Coefficient	---	---
		P-Value	---	---
H7I	0.9398	Coefficient	---	---
		P-Value	---	---
H8I	0.9398	Coefficient	---	---
		P-Value	---	---
H9I	0.9384	Coefficient	1.28	0.97
		P-Value	0.04	0.396
H10I	0.9368	Coefficient	1.23	1.11
		P-Value	0.064	0.414

material produced and total distance traveled provided the best fit of the eight forms tested regardless of whether or not the intercept was included in the model hypothesis. When those totals were partitioned by tether status or material type, the coefficients for those subgroups proved nearly identical to the total and each other in all cases. This suggests that the harvester was largely insensitive to both material size and tether status given the conditions on our site. This, in turn, implies that tethering on the harvester may function as a fixed cost, incurring delays due to the physical act of tethering and untethering, but not consistently slowing travel or material processing to a degree we could detect.

The strongest forwarder results required more complex models (Table 2.7 and Table 2.8). Partitioning distance traveled into tethered and untethered distance and total material weight into saw log weight and feedstock weight both improved model performance as measured by adjusted R-squared and AIC_C score (Table 2.9). The coefficients for these subgroups proved notably different as well, with tethered travel taking more time than untethered travel and feedstock logs taking more time than saw logs. Partitioning saw and feedstock collection by tether status also provided notably different coefficients, but resulted in lower adjusted R-squared values and higher AIC_C scores. The relatively large differences in tethered and untethered coefficient values suggests that tethering on the forwarder may function as both a fixed and variable cost, incurring not just the delay required to tether and untether, but also reducing the average speed of travel and material processing while tethered.

The differences in AIC_C scores (Δ_i) failed to provide convincing evidence for any model on either machine, suggesting that if several independent data sets were collected, they might have different best models (Burnham and Anderson, 2002). The AIC_C scores also failed to provide convincing evidence for, or against, our no-intercept assumption. Burnham and Anderson (2002) suggested that nested models with $\Delta_i < 2$ have substantial empirical support, i.e., that it is entirely plausible for the model with the inferior score to be the best model of the group, and nested models with $\Delta_i > 10$ have so little support that they could be omitted from future consideration. Only two model pairs produced $\Delta_i > 2$, F3 and F3I ($\Delta_i = 2.24$) and F7 and F7I ($\Delta_i = 2.32$). In addition, while the AIC_C scores for the forwarder favored the form without an

Table 2.7: Summary of base forwarder models

Model	Adj-R^2	Variable	ctot	utot	ttot	cdis	udis	tdis	csaw	usaw	tsaw	cbf	ubf	tbf
		β	1	2	3	4	5	6	7	8	9	10	11	12
		Units	gt	gt	gt	m	m	m	gt	gt	gt	gt	gt	gt
F3	0.9484	Coefficient	2.14	---	---	0.05	---	---	---	---	---	---	---	---
		P-Value	<0.001	---	---	<0.001	---	---	---	---	---	---	---	---
F4	0.9505	Coefficient	---	2.25	2.78	0.03	---	---	---	---	---	---	---	---
		P-Value	---	<0.001	<0.001	0.019	---	---	---	---	---	---	---	---
F5	0.9525	Coefficient	2.39	---	---	---	0.02	0.05	---	---	---	---	---	---
		P-Value	<0.001	---	---	---	0.121	<0.001	---	---	---	---	---	---
F6	0.9518	Coefficient	---	2.44	2.03	---	0.02	0.07	---	---	---	---	---	---
		P-Value	---	<0.001	0.005	---	0.173	0.015	---	---	---	---	---	---
F7	0.9506	Coefficient	---	---	---	0.04	---	---	1.95	---	---	3.31	---	---
		P-Value	---	---	---	0.003	---	---	<0.001	---	---	<0.001	---	---
F8	0.9565	Coefficient	---	---	---	---	0.01	0.05	2.19	---	---	4.06	---	---
		P-Value	---	---	---	---	0.613	<0.001	<0.001	---	---	<0.001	---	---
F9	0.9551	Coefficient	---	---	---	0.02	---	---	---	2.13	1.98	---	3.41	7.56
		P-Value	---	---	---	0.141	---	---	---	<0.001	0.002	---	<0.001	<0.001
F10	0.9567	Coefficient	---	---	---	---	0.01	0.06	---	2.29	1.31	---	4	5.94
		P-Value	---	---	---	---	0.724	0.032	---	<0.001	0.075	---	<0.001	0.013

Table 2.8: Summary of forwarder models with intercept

Model	Adj-R^2	Variable	intercept	ctot	utot	ttot	cdis	udis	tdis	csaw	usaw	tsaw	cbf
		β	0	1	2	3	4	5	6	7	8	9	10
		Units	NA	gt	gt	gt	m	m	m	gt	gt	gt	gt
F3I	0.9072	Coefficient	1.72	2.11	---	---	0.05	---	---	---	---	---	---
		P-Value	0.775	<0.001	---	---	<0.001	---	---	---	---	---	---
F4I	0.9128	Coefficient	7.66	---	2.17	2.9	0.03	---	---	---	---	---	---
		P-Value	0.241	---	<0.001	<0.001	0.054	---	---	---	---	---	---
F5I	0.9157	Coefficient	6.93	2.34	---	---	---	0.02	0.06	---	---	---	---
		P-Value	0.258	<0.001	---	---	---	0.228	<0.001	---	---	---	---
F6I	0.914	Coefficient	6.48	---	2.36	2.2	---	0.02	0.06	---	---	---	---
		P-Value	0.322	---	<0.001	0.003	---	0.252	0.037	---	---	---	---
F7I	0.9104	Coefficient	1.94	---	---	---	0.04	---	---	1.92	---	---	3.29
		P-Value	0.742	---	---	---	0.004	---	---	<0.001	---	---	<0.001
F8I	0.9242	Coefficient	8.62	---	---	---	---	0.001	0.05	2.11	---	---	4.11
		P-Value	0.143	---	---	---	---	0.95	<0.001	<0.001	---	---	<0.001
F9I	0.921	Coefficient	8	---	---	---	0.02	---	---	---	2.04	2.14	---
		P-Value	0.203	---	---	---	0.303	---	---	---	<0.001	0.001	---
F10I	0.9233	Coefficient	7.12	---	---	---	---	0.002	0.05	---	2.2	1.5	---
		P-Value	0.252	---	---	---	---	0.919	0.067	---	<0.001	0.047	---

Table 2.8: Summary of forwarder models with intercept (Continued)

Model	Adj-R^2	Variable	ubf	tbf
		β	11	12
		Units	gt	gt
F3I	0.9072	Coefficient	---	---
		P-Value	---	---
F4I	0.9128	Coefficient	---	---
		P-Value	---	---
F5I	0.9157	Coefficient	---	---
		P-Value	---	---
F6I	0.914	Coefficient	---	---
		P-Value	---	---
F7I	0.9104	Coefficient	---	---
		P-Value	---	---
F8I	0.9242	Coefficient	---	---
		P-Value	---	---
F9I	0.921	Coefficient	3.43	7.52
		P-Value	<0.001	<0.001
F10I	0.9233	Coefficient	3.98	6
		P-Value	<0.001	0.012

Table 2.9: Relative model strength by R^2 and second order Akaike's Information Criterion (AIC_C)

	Harvester Models					Forwarder Models				
Model	Adj-R^2	AIC_C	Δ_i^a	w_i^b	ER-H3[c]	Adj-R^2	AIC_C	Δ_i^a	w_i^b	ER-F8[d]
M3	0.9715	423.28	0	0.2137	---	0.9484	531.03	6.05	0.0146	20.59
M3I	0.9412	423.31	0.03	0.2105	1.02	0.9072	533.27	8.29	0.0048	63.12
M4	0.971	425.49	2.21	0.0708	3.02	0.9505	530.36	5.38	0.0205	14.73
M4I	0.9412	424.74	1.46	0.1030	2.08	0.9128	531.29	6.31	0.0129	23.45
M5	0.971	425.51	2.23	0.0701	3.05	0.9525	528.41	3.43	0.0542	5.56
M5I	0.9412	424.8	1.52	0.0999	2.14	0.9157	529.45	4.47	0.0323	9.35
M6	0.9703	428.03	4.75	0.0199	10.75	0.9518	530.49	5.51	0.0192	15.72
M6I	0.9399	427.36	4.08	0.0278	7.69	0.914	531.94	6.96	0.0093	32.46
M7	0.9708	425.7	2.42	0.0637	3.35	0.9506	530.43	5.45	0.0198	15.26
M7I	0.9398	425.85	2.57	0.0591	3.61	0.9104	532.75	7.77	0.0062	48.67
M8	0.9703	428.04	4.76	0.0198	10.80	0.9565	524.98	0	0.3014	---
M8I	0.9398	427.43	4.15	0.0268	7.96	0.9242	525.12	0.14	0.2811	1.07
M9	0.9695	430.68	7.40	0.0053	40.45	0.9551	528.08	3.1	0.0640	4.71
M9I	0.9384	430.14	6.86	0.0069	30.88	0.9210	528.89	3.91	0.0427	7.06
M10	0.9687	433.49	10.21	0.0013	164.84	0.9567	527.73	2.75	0.0762	3.96
M10I	0.9368	433.07	9.79	0.0016	133.62	0.9233	528.97	3.99	0.0410	7.35

[a] The absolute difference in AICc scores between a given model and the best model

[b] Akaike's weights, the likelihood that a given model is correct for this data and set of models

[c] The evidence ratio, how many times more likely it is that H3 is correct for this data when compared to a given model

[d] The evidence ratio, how many times more likely it is that F8 is correct for this data when compared to a given model

intercept in every case, the AIC_C scores for the harvester favored the form with an intercept six times out of eight.

The idea that the models are all reasonably effective for the data can be further supported by the similarity of their predictions (Table 2.10). When supplied with our observed average distances and tonnage for a single hectare we find estimated treatment times ranging from 433 to 440 productive minutes for the harvester (with an average of 437 and standard deviation of 2.4) and 433 to 464 productive minutes for the forwarder (with an average of 447 and standard deviation of 12.1). If we then calculate the average stump-to-truck cost for our high cost machine rate scenario we find values ranging from $24.93 per green metric ton to $25.78 per green metric ton (with an average of $25.35 and standard deviation of $0.28).

Table 2.10: Estimated machine time in productive machine minutes (PMM), machine cost in dollars per hectare, and stump-to-truck costs in dollars per green metric ton (GT) for one hectare treated, by model form.

	Estimated Time		Estimated Cost		
Model	Harvester (PMM)	Forwarder (PMM)	Harvester ($ per Ha)	Forwarder ($ per Ha)	Stump-to-truck ($ per GT)
M1	440.26	464.51	1772.63	1400.39	25.78
M2	439.00	433.12	1767.55	1305.74	25.09
M3	434.23	432.25	1748.34	1303.12	24.93
M4	437.97	444.83	1763.40	1341.05	25.30
M5	439.08	457.35	1767.86	1378.79	25.60
M6	433.29	451.98	1744.57	1362.60	25.32
M7	436.77	440.38	1758.56	1327.62	25.18
M8	436.69	458.45	1758.26	1382.11	25.56

Model Use

While the models presented here fit the data relatively well, potential users must keep two critical facts in mind to avoid overstating model effectiveness or returning unreasonable results. First: the study did not observe, and therefore cannot claim to describe, extreme slope operations beyond the limits observed in this study. While isolated slopes on the study sites reached up to 70% slope, steeper operations have been observed in other regions and we cannot project that machine behavior will remain consistent on those extreme slopes. There may be a slope threshold beyond which the time required for any given action is significantly increased due to increased difficulty of maneuvering and material handling, or reduced operator confidence. Second: the models in this study use total distance traveled, not average yarding distance.

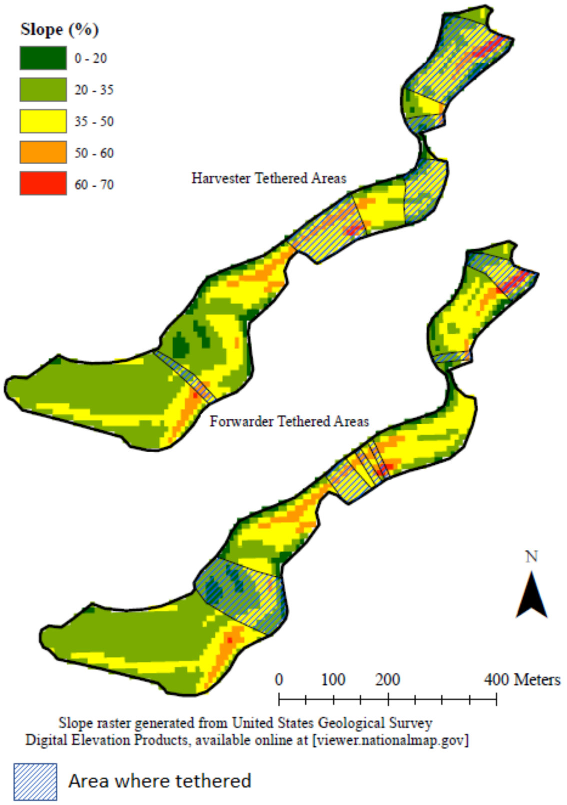
For planning purposes, distance traveled for the harvester can be estimated as a function of the linear distance required to cover an area, given an average length of boom extension. For example, if we assume a ten-meter maximum reach and an average boom extension of 80%, this would result in a cutting swath sixteen meters wide, requiring 556 meters of linear distance for that swath to cover one hectare. As a result of this assumption, the average in-stand distance traveled by the harvester to treat a single hectare should be between 556 meters (on shallow slopes where the harvester can avoid driving back on its own trail) and 1,112 meters (on steep slopes where the harvester must go out and back on a single trail with no opportunistic wandering). It is important to note that this estimate does not include the distance required to move from corridor to corridor, which could significantly underestimate the true distance traveled for stands with short average yarding distances. During this study, we observed an average harvester distance traveled of 943 meters per hectare.

Distance traveled for the forwarder can be estimated as a function of average yarding distance, material loading, and bunk utilization. The weight of material to be gathered (in tonnes per hectare) can be divided by an assumed average bunk weight (in tonnes per turn) and rounded to produce a number of turns per hectare. The number of turns per hectare can be multiplied by the average yarding distance to produce an estimate of the total forwarder distance traveled per hectare. This should be subject to a minimum distance traveled per hectare, calculated as a function of boom reach (as described above) to avoid underestimating the distance traveled when very small quantities of material are being removed. As with the harvester distance estimate, this does not include the distance required to move from corridor to corridor and may underestimate the true distance traveled when average yarding distances are short. During this study we observed an average forwarder distance traveled of 3,315 meters per hectare with an average of twelve bunks per hectare.

General Observations and Limits on Inference

The study was designed with the goal of collecting data while interfering in operations as little as possible, and we largely succeeded at that goal. However, it also identified aspects of cut-to-length systems that make them relatively difficult to study, as well as peculiarities of the study area itself that limit the ability to draw broad inference from the results. Unlike cable logging, corridor boundaries with ground-based systems are more porous. Both machines tended to wander opportunistically while untethered, a behavior Flint (2014) observed in his flat ground unit. On our study, even while tethered it was not uncommon for the forwarder to gather or shovel material from one corridor while in another, particularly along the tops of ridges and the bottoms of valleys where corridors often converge, overlap, or terminate in close proximity to each other. This porosity is what necessitated the consolidation of some corridors for this analysis.

In addition, variability in soil conditions prevented this study from being able to make any confident inferences on the general effect of slope on operations. The interaction between soil strength and ground slope on the limits of off-road vehicle performance is well known (e.g. Visser and Stampfer 2015, Sessions et al. 2017). Both operators commented that the weak soils on this site drove the decision to tether at flatter slopes than they were accustomed to in western Oregon. Both operators expressed concerns over getting bogged down and/or causing undue damage to the soil, and made the decision to use the tether to address those concerns (C. Cano and J. Vidrio, personal communication, 2016). This subjective operator judgement call on a corridor-by-corridor and bunk-by-bunk basis may also explain the notable differences in when and where each machine used their tether (Figure 2.4: Areas of tethered and untethered operation by machine).



Slope raster generated from United States Geological Survey
Digital Elevation Products, available online at [viewer.nationalmap.gov]

Figure 2.4: Areas of tethered and untethered operation by machine

Lastly, while our results indicate that tethering on a harvester behaves like a fixed cost and tethering on a forwarder behaves like a fixed and variable cost, it is possible that this behavior is actually an operator effect and not a machine effect. Operator effects have long been known to have a significant and difficult to quantify effect on machine productivity (Gullberg 1995), and our use of a single operator per machine does not allow a means of identifying the difference between a machine effect and an operator effect.

Need for Future Work

This case study serves as a starting point for examining the potential of these tethered cut-to-length systems, but more work is required. The conditions of the site and treatment did not allow us to study the effects of extreme slopes, yarding distances, haul direction, tree retention density, or operator experience with any level of rigor. A more controlled study, with a wider variance in site conditions could do a great deal to corroborate the results found here.

Acknowledgements

This study was funded by the Institute for Working Forest Landscapes as a part of the "Opportunities for biochar production to reduce forest wildfire hazard, sequester carbon, and increase agricultural productivity of dryland soils" project grant. The authors would like to thank Chuey Cano, Jose Vidrio, and Matt Mattioda of Miller Timber for all their help and patience throughout the course of this study, as well as Paul Cheng, Judd Lehman, Regina Rone, and Chaylon Shuffield of the Fremont-Winema National Forest for their input and support, and Ariel Muldoon, Chu Qi, and Bogdan Strimbu of Oregon State University for statistical and drone support.

References

Amishev, D., Evanson, T. and Raymond, K., 2009. Felling and Bunching on Steep Terrain–A Review of the Literature. FFR Technical Note, 1(7), pp.1-10.

Brinker, R.W., Kinard, J., Rummer, R., and Lanford, B.L., 2002. Machine rates for selected forest harvesting machines.

Burnham, K.P. and Anderson, D.R., 2002. Model selection and multimodel inference: a practical information-theoretic approach. Springer.

Christian, J.A. 2014. Oregon Department of Forestry logging cost update. Master of Forestry project, Department of Forest Engineering, Resources, and Management, Oregon State University, Corvallis, Oregon.

Eisenhauer, J.G., 2003. Regression through the origin. Teaching statistics, 25(3), pp.76-80.

Flint, B.R., 2013. Analysis and operational considerations of biomass extraction on steep terrain in western Oregon. Master of Science, Oregon State University, Corvallis, OR.

Fonseca, M.A., 2005. The measurement of roundwood: methodologies and conversion ratios. CABI.

Ghaffariyan, M.R., Acuna, M. and Ackerman, P., 2012. Review of new ground-based logging technologies for steep terrain. CRC for Forestry Bulletin, pp.1-5.

Gullberg, T., 1995. Evaluating operator-machine interactions in comparative time studies. Journal of Forest Engineering, 7(1), pp.51-61.

Jain, T.B., Battaglia, M.A., Han, H.S., Graham, R.T., Keyes, C.R., Fried, J.S. and Sandquist, J.E., 2012. A comprehensive guide to fuel management practices for dry mixed conifer forests in the northwestern United States.

Keegan III, C.E., Morgan, T.A., Blatner, K.A. and Daniels, J.M., 2010. Trends in lumber processing in the Western United States. Part I: Board foot Scribner volume per cubic foot of timber. Forest products journal, 60(2), pp.133-139.

Kellogg, L.D. and Bettinger, P., 1994. Thinning productivity and cost for a mechanized cut-to-length system in the Northwest Pacific Coast region of the USA. Journal of Forest Engineering, 5(2), pp.43-54.

Lindroos, O. and Cavalli, R., 2016. Cable yarding productivity models: a systematic review over the period 2000–2011. International Journal of Forest Engineering, 27(2), pp.79-94.

Lotan, J. 1976. Cone serotiny - fire relationships in lodgepole pine. *In*: Tall Timbers Fire Ecology Conference Proceedings 14, Tall Timbers Research Center, Tallahassee, FL. pp. 267-278

McIver, J.D., Adams, P.W., Doyal, J.A., Drews, E.S., Hartsough, B.R., Kellogg, L.D., Niwa, C.G., Ottmar, R., Peck, R., Taratoot, M. and Torgersen, T., 2003. Environmental effects and economics of mechanized logging for fuel reduction in northeastern Oregon mixed-conifer stands. Western Journal of Applied Forestry, 18(4), pp.238-249.

Motulsky, H. and Christopoulos, A., 2004. Fitting models to biological data using linear and nonlinear regression: a practical guide to curve fitting. Oxford University Press.

North, M., Brough, A., Long, J., Collins, B., Bowden, P., Yasuda, D., Miller, J. and Sugihara, N., 2015. Constraints on mechanized treatment significantly limit mechanical fuels reduction extent in the Sierra Nevada. Journal of Forestry, 113(1), pp.40-48.

Oregon Department of Forestry. 2017. Yearly pond values [online]. Available from http://www.oregon.gov/ODF/Working/pages/TimberSales.aspx [accessed 27 July, 2017]

Ryan, K.C., Knapp, E.E. and Varner, J.M., 2013. Prescribed fire in North American forests and woodlands: history, current practice, and challenges. Frontiers in Ecology and the Environment, 11(s1).

Sessions, J., Leshchinsky, B., Chung, W., Boston, K., and Wimer, J. 2017. Theoretical Stability and Traction of Steep Slope Tethered Feller-Bunchers. Forest Science 63(2):192-200.

Stephens, S.L. and Ruth, L.W., 2005. Federal Forest-Fire Policy in the United States. Ecological applications, 15(2), pp.532-542.

United States Department of Agriculture. 2017. OR.accdb. Available from https://apps.fs.usda.gov/fia/datamart/datamart_access.html [accessed 10 May, 2017]

Visser, R. and Stampfer, K., 2015. Expanding ground-based harvesting onto steep terrain: a review. Croatian journal of forest engineering, 36(2), pp.321-331

EVALUATING BIOCHAR FEEDSTOCK SUPPLY POTENTIAL AND THE EFFECTS OF AN OPERATING BIOCHAR FACILITY ON THE FEASIBILITY OF FEDERAL FUELS REDUCTION TREATMENTS IN THE UPPER KLAMATH BASIN

Joshua H. Petitmermet, Jeremy S. Fried, John Sessions

Abstract

Fuels reduction treatments are an effective way of reducing fire hazard on the federal forests of the western United States, but are often considered economically infeasible due to high treatment costs and poor markets for small diameter material. Biochar production, combined with tethered cut-to-length technology, has the potential to improve the economic feasibility of treatments by providing a new outlet for the material produced. To assess the potential for fuels treatment outputs to supply a biochar processing facility we performed a wood basket analysis of the federal forested lands of the Upper Klamath Basin in south-central Oregon and northern California. A variety of fuels treatments were modeled and applied to the landscape over a twenty-year planning horizon, and the effects of four different outcomes for biochar feedstock and two levels of potential subsidy on individual treatment feasibility were evaluated. We found that federal fuels treatments were capable of supplying between 91 and 241 thousand bone dry tonnes annually with availability driven primarily by limits on the number of hectares treated. For the first 45,360 bone dry tonnes produced annually average costs ranged from $39 to $42 per bone dry metric ton and marginal costs ranged from $49 to $56 dollars per bone dry metric ton, assuming the cost of forwarding, loading, and truck transport are all included.

Introduction

Effective, long-term resource management must meet three criteria to be successful: it must be biologically possible, culturally adoptable (socially acceptable), and economically feasible (Firey 1960). For fuels reduction treatments on federal lands in the state of Oregon, the first two are rarely a concern. By their nature, fuels reductions treatments seek to modify stand structures and fuel loading, not to maximize production, and mechanical fuels treatments have found greater than majority support in Oregon (Brunson and Shindler, 2004) with even greater support along the wildland urban interface (Toman et al. 2011). Economic feasibility, however, can be difficult to determine as it is contingent on site specific and market factors. This is particularly true of the Upper Klamath Basin in south-central Oregon and northern California where the lack of local markets for pulpwood or any other small diameter material often elevates fuel treatment costs beyond what is affordable by necessitating some kind of *in situ* treatment of unmerchantable wood, such as via pile and burn.

Petitmermet et al. (in preparation) identified two emerging technologies with the potential to change the economics of mechanical fuels treatments: tethered cut-to-length systems that may reduce stump-to-truck costs, and biochar, a forest product well-suited to utilizing small diameter wood. Together, these could reduce treatment costs and increase the net revenue generated by fuels treatments. That study examined the stump-to-truck costs of a steep slope fuels treatment implemented with tethered cut-to-length systems. This study seeks to identify the potential for a wood basket comprised of the federally-owned forested acres of the Upper Klamath Basin to support a hypothetical biochar processing facility through fuels reduction treatments aimed at increasing fire resistance and resilience across the landscape.

In this analysis we assumed the construction of a biochar facility on a brownfield site at Worden, Oregon that would be capable of processing 45,360 bone dry tonnes (BDT) of woody material annually over a twenty-year lifespan. David Smith and Will Holloman of Oregon State University selected the site and defined the facility characteristics as part of a related study (Sessions et al. 2018). This study assessed the potential of the wood basket to generate the quantity of material the facility would require, the average and marginal cost of the material at the facility gate, and the effect of payments for biochar feedstock and per acre subsidies on both the economic feasibility of individual treatments, and effect of biochar feedstock payments on the allocation of treatments over a twenty-year planning horizon.

Site Description

The Upper Klamath Basin spans 2.1 million hectares, including more than one million hectares within the boundaries of seven different national forests: Deschutes, Fremont-Winema, Klamath, Modoc, Rogue River-Siskiyou, Shasta-Trinity, and Umpqua. The majority of the forested area is classified by the Wildland Fire Science, Earth Resources Observation and Science Center (2016) as being in fire regime group one: characterized by high frequency (0-35 year return intervals) and low/mixed severity fire (Barrett et al. 2010). On the 931,000 hectare Fremont-Winema, approximately 4,050 hectares per year were harvested, pre-commercially thinned, and/or influenced by a prescribed fire between 2007 and 2016 (P. Cheng, personal communication, 2017). If we assume that half the area treated with prescribed fire also receives mechanical treatment, there would be 10,120 hectares of management activity per year occurring on the Fremont-Winema, implying a management return interval of approximately 100 years.

Methods

Overview

This study builds on a recent study of the cost and productivity of tethered and untethered cut-to-length systems in dry forest fuels reduction treatments in south-central Oregon (Petitmermet et al. in preparation). All cycle time models (used to estimate the amount of time required to process a given amount of material) and cost parameters used in this analysis are derived from that study. All harvester and forwarder costs were generated using the “high cost” machine rate scenario described there with harvester utilization rates of 64% and 87% for tethered and untethered operations respectively and forwarder utilization rates of 76% and 89% for tethered and untethered operations respectively.

To select, assess, and optimize the effects of multiple treatments across such a large area we used a process that combined the Bioregional Inventory Originated Simulation Under

Management (BioSum, Fried et al. 2017a) and a purpose-built heuristic program (Neo-Processor). The merits of any potential treatment were measured by comparing the composite fire resistance score (CRS, Fried et al. 2017b) of the potential treatment to a no-action alternative, and maximizing that improvement subject to economic constraints. The potential effects of developing markets were simulated via four market scenarios designed to explore different prices and outcomes for biochar feedstock, and three different levels of potential treatment capacity on the part of the Forest Service. All modeling is done over a twenty-year planning horizon using four, five-year periods. No reentry was allowed on any acre over the planning horizon, so each acre could be treated, at most, once.

BioSum – A Brief History

For over a decade researchers with the Pacific Northwest Research Station of the United States Forest Service have been developing, updating, and (more recently) distributing the Bioregional Inventory Originated Simulation Under Management (BioSum), an analysis framework created to evaluate the effectiveness and feasibility of large numbers of candidate silvicultural operations, including fuel treatments, on multimillion hectare forested landscapes (Fried et al. 2017a). BioSum relies on publicly available forest inventory data collected by the Forest Service's Forest Inventory and Analysis (FIA) program as a source of stand and tree level data for a "test bed" on which users can define and simulate silvicultural prescriptions using the Forest Vegetation Simulator (FVS), and ultimately optimize treatment selection to, for example, maximize net or gross revenue, wood volume generated, or long-term treatment effectiveness. BioSum software and accompanying documentation is available at no cost from www.biosum.info.

We relied on the BioSum framework and software to accomplish three key tasks in this analysis: to guide the assembly of an inventory database for our study area, to estimate haul cost and average yarding distance, and to model the outcomes of alternative treatments in FVS.

BioSum - Translating FIA Data to FVS

Forest data for the states of Oregon and California were downloaded from the FIA DataMart website (https://apps.fs.usda.gov/fia/datamart/) on May 22nd, 2017. The BioSum Database tools were then used to subset the data on two criteria:

- Being situated within the boundaries of the Upper Klamath basin of south-central Oregon and northern California, and
- Containing at least one unreserved, forested condition class under federal ownership.

These 871 full or partial plots, representing nearly 769,000 hectares (a third of the total project area) served as a basis for the BioSum project dataset used for analysis. BioSum converted these data into FVS-readable format to make them ready for simulating fuel treatment and stand projection.

BioSum - Estimating Haul Cost and Yarding Distance

While BioSum is predominantly a non-spatial framework, it does require spatial data for the calculation of haul costs and yarding distances. For this study, the spatial data were assembled from three sources. Boundary and transportation data were retrieved via the USGS National Map Viewer (https://viewer.nationalmap.gov/basic/), specifically the:

- USGS National Boundary Dataset for California 20170424
- USGS National Transportation Dataset for California 20170327, and the
- USGS National Transportation Dataset for Oregon 20170326

“Fuzzed” plot locations were obtained from each states’ FIADB database. A list of currently operational sawmills in the region, along with the Worden, OR candidate site for a biochar processing facility, served as final destinations for harvested wood generated by the simulations in this analysis (Table 3.1). All spatial information processing was performed in ArcMap 10.4.1 or ArcCatalog 10.4.1.

Trucking costs were estimated as a function of speed, load status, and state of operation (Table 3.2). The weight of truck loads that would pass through California at any point *en route* to

Table 3.1: Regional processing site locations, products produced and feedstock

Site	Products	Feedstock	City	State
Potential Worden Site	Biochar	Biochar Feedstock	Worden	OR
Columbia Forest Products Plywood	Veneer/ Plywood	Saw Logs	Klamath Falls	OR
Jeld-Wen Thomas Lbr. Sawmill	Lumber	Saw Logs	Klamath Falls	OR
Roseburg Forest Products Veneer Mill	Veneer	Saw Logs	Weed	CA
Boise Wood Products Veneer Mill	Veneer	Saw Logs	White City	OR
Murphy Softwood Veneer Plant	Veneer	Saw Logs	White City	OR
Fruit Growers Supply Co. Sawmill	Lumber	Saw Logs	Yreka	CA
Timber Products Veneer Mill	Veneer	Saw Logs	Yreka	CA
Collins Company - Lakeview Sawmill	Lumber	Saw Logs	Lakeview	OR

Table 3.2: Estimated hourly truck transport costs for saw logs and biochar feedstock

Speed (kilometers per hour)	89	56	40
Loaded - Oregon	$126	$113	$106
Loaded - California	$114	$101	$94
Unloaded - Oregon	$99	$86	$79
Unloaded - California	$95	$82	$75

processing facilities was assumed to be 21 green tonnes, versus 28 green tonnes for those that remained in Oregon. This assumption could lead to an underestimate of haul costs given that load size may be limited by the volumetric capacity of the truck before a state's vehicle gross weight limit. These unit costs and load assumptions allowed calculation of round-trip (i.e., accounting for back-haul) costs per green ton per one-way hour (Table 3.3).

Table 3.3: Round trip cost per green metric ton for transport of saw log and biochar feedstock ($ per one-way hour[a]), by rated road speed.

Speed (kilometers per hour)	89	56	40
Oregon	$8.00	$7.05	$6.58
California	$10.97	$9.57	$8.87

[a] For example, a 4-hour round trip to Worden, Oregon at 56 kilometers per hour would cost $14.10 per green metric ton (2 x $7.05)

The National Transportation Database information for each state was clipped to an area extending 50 kilometers from the project area boundary, then merged into a single road network file. Average speeds were assigned to each road segment based on its MTFCC (MAF/TIGER Feature Class Code) classification and used to calculate a one-way travel time. The one-way travel time for each road segment was then multiplied by the round-trip cost per green ton (in dollars per one-way hour) to populate each road with two costs, one appropriate for loads that originate in or pass through California at any point ("CA Cost"), and one appropriate for loads that begin, transit through, and terminate in Oregon alone ("OR Cost").

We created three "New Closest Facility" datasets with the network analyst tool:

- CAT: The optimal route from each plot in the project area to each facility in California, accumulating CA Cost
- ORCAT: The optimal route from each plot in California to each facility in Oregon, accumulating CA Cost, and

- ORORT: The optimal route from each plot in Oregon to each facility in Oregon, accumulating ORCost.

The results for the three datasets were then combined into a single haul cost database describing the haul cost in dollars per green ton to move each kind of material from any plot to any processing facility of the appropriate type.

To ensure that each site and plot has access to the road network for analysis, nodes in the plot layer were moved to the nearest road using ArcMap's "Near" function, with movement distance saved for use as the average yarding distance input to treatment cost calculations.

BioSum - Prescription Modeling

Six types of treatment were devised with the intent of providing the analysis with a range of feasible, effective prescriptions with varying levels of removal. All treatments fall into one of three categories:

- Thins-from-below
- Q-factor treatments, and
- Pseudo-clearcuts

All prescriptions were coded as FVS Keyword Control Programs (KCP) and implemented via the Southern Oregon (SO) variant of FVS (version 1778) using Suppose 2.06. All prescriptions were simulated over a twenty-year planning horizon as four, five-year cycles on all stands in the project area, with a triggering basal area coded into each treatment to ensure that treatments could only run on well stocked or overstocked stands. To avoid simulating thinning operations that would remove unrealistically small amounts of material, both the thin-from-below and q-factor treatments utilize a "proportional removal requirement." This requirement calculates how much material an entry could remove, and only allows treatment if the potential quantity removed was proportionally greater than or equal to the minimum possible thin. For example, if the thin was triggered by reaching (or exceeding) 30 square meters of basal area per hectare and the thinning target was 20 square meters of basal area per hectare, treatment would only be allowed if it removed at least one third of the stand's total basal area. This requirement, combined with the diameter caps associated with each prescription (Table 4), effectively prevented treatment of stands with a large proportion of basal area in trees larger than the diameter limit.

Table 3.4: Summary of requirements, basal area targets, and maximum take tree diameter at breast height (DBH) for all treatments simulated.

		Treatment requirements				
Treatment style	Treatment type	Dominant species[a]	Strata count[b]	Minimum basal area (m^2 per ha)	Residual basal area target (m^2 per ha)	Maximum DBH of take trees
Thin from below	1	Any or none	1	34.4	23.0	25.4 cm
	2	Any or none	1	34.4	23.0	40.6 cm
	3	Any or none	1	27.5	17.2	53.3 cm
Q-factor	4	Any or none	>= 2	28.7	17.2	50.8 cm
	5	Any or none	>= 2	25.3	11.5	61 cm
Pseudo-clearcut	6	PICO[c]	Any	18.4	N/A[e]	N/A
		JUOC[d]	Any	8.0	N/A[e]	N/A

Commented [JS2]: Are the units correct?

[a] A species is considered dominant if it comprises 70% or more of the basal area for the stand
[b] Strata count taken from Forest Vegetation Simulator STRCLASS output tables
[c] Lodgepole pine, *Pinus contorta*
[d] Western juniper, *Juniperus occidentalis*
[e] Type six (pseudo-clearcut) treatments remove all trees of the dominant species, and leave all others

Up to twelve thin-from-below treatments were modeled for each single-story stand. Treatments were differentiated by intensity, with respect to residual basal area and diameter limit (Table 4), and period in which they were implemented. These treatments were implemented in FVS using the THINBBA keyword and a species removal preference. Species preferences for THINBBA treatments “weight” the removal priority of each tree in a given species group, such that those with higher weights are selected first. For our purposes, the “priority value” (where larger values translate to “cut these trees first”) for a trees removal by THINBBA is calculated as:

(1) Priority = (-1 * DBH) + W

Where W is the weight specified. Western juniper and lodgepole pine were assigned a weight of fifty, true firs, Douglas-fir, and incense-cedar a weight of twenty-five, and all other species a weight of zero.

Multi-storied stands, defined as having more than one stratum listed in the FVS Stratum table using the default criteria for strata determination, were assigned a specific q-factor as a function of their dominant species. A species (or species group, in the case of true fir) was considered dominant if it comprised more than seventy percent of the total stand basal area.

Eight q-factor treatments were modeled, describing two treatments of differing intensity across each of the four planning periods. Q-factor treatments use the same species preferences as thin from below treatments. All q-factor thins were performed using the THINQFA keyword in FVS and all stands were assigned a specific q-factor as a function of their dominant species (Table 3.5).

Table 3.5: Q-factors by dominant species

Dominant species[a]	Q-factor
Ponderosa pine	1.15
Incense-cedar	1.25
True fir	1.35
Douglas-fir	1.4
None	1.3

[a] A species is considered dominant if it comprises 70% or more of the basal area for the stand

Four pseudo-clearcut treatments were modeled, with one treatment in each of the four planning periods. Pseudo-clearcut treatments were only allowed on stands where lodgepole pine or western juniper comprised seventy percent or more of the basal area and a basal area target was met (18.4 square meters per hectare for lodgepole, 8 square meters per hectare for juniper). If both conditions were met,

- The non-lodgepole, non-juniper basal area was calculated as "_XXB"

- Lodgepole and juniper were assigned a species preference of 99, with all others assigned a species preference of 0, and
- A THINBBA treatment was scheduled to thin the stand down to a basal area target of "_XXB"

The end result of this is that in overstocked lodgepole and juniper stands, all lodgepole and juniper were removed and any trees of other species were retained.

Any given stand was considered eligible for any specific treatment only if all criteria for that treatment are met. For thin-from-below and q-factor treatments this includes the minimum basal area requirement and the proportional removal requirement, and for the pseudo-clearcuts this includes the minimum basal area requirement and the species composition requirement.

Neo-Processor - General Purpose

The BioSum software provides a compelling framework for work flow automation, with the capability of pricing wood volume by tree size and diameter class, but it does not provide for merchandising individual trees into multiple products and individual log sorts and prices. Because there is significant biochar feedstock within most trees that also produce merchantable wood, and it was important to differentiate this, we designed an alternate optimization module to account for wood product volumes and values produced, that we dubbed Neo-Processor. Neo-Processor offers three key advantages over the Processor module in BioSum: optimal bucking of cut trees in the FVS output, ability to price each individual log made from those bucked cut trees according to local market values, and the ability to accurately assess and opportunistically optimize the amount of biochar feedstock material generated.

For the sake of brevity, we describe what Neo-Processor does, not how it works. Interested readers can find the complete Neo-Processor program and documentation, including a full description of the code and a copy of the scripts with sample inputs and outputs on Github at github.com/JPetitmermet/Neo-Processor. The module itself is composed of two separate scripts written in the Python 2.7 programming language and executed in the Spyder integrated development environment: the Evaluator, which determines revenues, costs, and material outputs, and the Optimizer, which uses those values to assign treatments and perform landscape level optimization.

Neo-Processor - Evaluator

The Evaluator module takes as input the FVS output (tree lists, cut lists, POTFIRE tables, and STR_CLASS tables) and GIS generated data (average yarding distances and route costs) to process through three submodules:

- Tree Merchandiser
- Accountant, and
- Fire Resistance Evaluator

Tree Merchandiser uses the FVS cut list for each prescription to evaluate the material outputs produced by that prescription. This includes:

- Optimally bucking each tree in the cut list capable of producing sawlog-quality material
- Calculating the volume of biochar feedstock produced by each tree in the cut list, and
- Using the outputs from that bucking and calculation to determine the expected yields per acre for each stand treated by that prescription.

Tree Merchandiser has six species codes (Table 3.6) comprised of twelve species, selected from the species present in the Upper Klamath basin. Species were included in Tree Merchandiser only if they met each of three criteria:

- Acceptable to local mills for producing traditional, high value wood products (not pulp or biochar)
- Likely to be removed during a treatment aimed at fuels reduction, and
- With available price and sort data.

Price and sort information in Tree Merchandiser is built to reflect the Klamath Unit average of the four most recent quarters available on the State of Oregon’s Open Data website at the time of Tree Merchandiser’s design, specifically quarters three and four of 2015 and quarters

Table 3.6: Tree Merchandiser species codes, aggregates, and commercial status

In FVS / FIA	
FIA Code	Common Name
🟩 11	Pacific silver fir
🟩 15	White fir
🟩 17	Grand fir
🟩 19	Subalpine fir
🟩 20	Red fir
🟩 21	Shasta fir
🟧 64	Western juniper
🟩 81	Incense cedar
🟧 93	Engelmann spruce
🟧 101	Whitebark pine
🟩 108	Lodgepole pine
🟩 116	Jeffrey pine
🟩 117	Sugar pine
🟧 119	Western white pine
🟩 122	Ponderosa pine
🟩 202	Douglas-fir
🟧 264	Mountain hemlock
🟧 321 / 998	Rocky mountain maple
🟧 431	Chinkapin
🟧 475	Mountain mahogany
🟧 746	Quaking aspen
🟧 763	Chokecherry
🟧 815	Oregon white oak
🟧 818	California black oak

Species Status for Bucking	
🟩	Commercial Species
🟧	Non-commercial Species

In Tree Merchandiser	
TM Code	Common Name
1	Douglas-fir
2	Incense cedar
3	Jeffrey pine
3	Ponderosa pine
4	Sugar pine
5	Lodgepole pine
6	Pacific silver fir
6	White fir
6	Grand fir
6	Subalpine fir
6	Red fir
6	Shasta fir

one and two of 2016 (Table 3.7). Five of the six Tree Merchandiser species are priced with the same diameter class breakdown, but lodgepole pine has only a single “camp run” value. A camp run log is any log of greater than cull quality (Bell and Dillworth, 1988).

Tree Merchandiser’s optimal bucking process uses a forward reaching algorithm (Denardo 2012) to find the best log or combination of logs that could come from any given merchantable tree to a minimum top inside diameter of 15.24 centimeters (six inches). Once the optimal set of logs for the tree is found, the model determines how much (if any) biochar feedstock can be produced from the remainder using only bole material to a minimum top inside diameter of 10.16 centimeters (four inches). Tree Merchandiser only allows saw logs of 2.4, 3.7,

Table 3.7: Average log prices by sort and species in dollars per thousand board feet (Eastside Scribner scale)

Species	Sort (inches)	Sort (cm)	2015.3	2015.4	2016.1	2016.2	Ave.
Douglas-fir	6-8	15.24-20.32	$465.00	$475.00	$490.00	$495.00	$481.25
	8-14	20.32-35.56	$510.00	$520.00	$525.00	$500.00	$513.75
	14-22	35.56-55.88	$520.00	$535.00	$530.00	$520.00	$526.25
	22+	55.88+	$525.00	$545.00	$535.00	$525.00	$532.50
Incense-cedar	6-8	15.24-20.32	$650.00	$600.00	$650.00	$640.00	$635.00
	8-14	20.32-35.56	$650.00	$600.00	$650.00	$640.00	$635.00
	14-22	35.56-55.88	$650.00	$600.00	$650.00	$640.00	$635.00
	22+	55.88+	$650.00	$600.00	$650.00	$640.00	$635.00
Ponderosa pine	6-8	15.24-20.32	$285.00	$270.00	$265.00	$310.00	$282.50
	8-14	20.32-35.56	$320.00	$315.00	$305.00	$325.00	$316.25
	14-22	35.56-55.88	$355.00	$350.00	$335.00	$340.00	$345.00
	22+	55.88+	$390.00	$385.00	$365.00	$370.00	$377.50
Sugar pine	6-8	15.24-20.32	$285.00	$280.00	$275.00	$260.00	$275.00
	8-14	20.32-35.56	$305.00	$295.00	$300.00	$280.00	$295.00
	14-22	35.56-55.88	$335.00	$360.00	$320.00	$305.00	$330.00
	22+	55.88+	$365.00	$335.00	$345.00	$335.00	$345.00
Lodgepole pine	CR[a]	CR[a]	$325.00	$320.00	$360.00	$365.00	$342.50
True fir[b]	6-8	15.24-20.32	$380.00	$385.00	$370.00	$375.00	$377.50
	8-14	20.32-35.56	$410.00	$390.00	$380.00	$385.00	$391.25
	14-22	35.56-55.88	$415.00	$405.00	$410.00	$410.00	$410.00
	22+	55.88+	$420.00	$415.00	$415.00	$415.00	$416.25

[a] Camp run (CR) logs are defined as any log of greater than cull quality (Bell and Dillworth, 1988)

[b] The true fir group includes all of the following species: Pacific silver fir, white fir, grand fir, subalpine fir, red fir, and Shasta fir

4.9, and 6.1 meters (eight, twelve, sixteen, and twenty feet) in length and assumes 0.3 additional meters (one foot) per log of trim and a 0.3-meter stump. Biochar feedstock is assumed to be log-like with no trim and can be of any length greater than eight feet. Unmerchantable trees (those without a Tree Merchandiser species code) are not bucked, and the entirety of their bole material (from the 0.3-meter-tall stump to a 10.16 centimeter inside bark diameter top) is used as biochar feedstock. As each cut tree is processed, Tree Merchandiser calculates the per acre revenue, saw-quality volume and green weight, and biochar feedstock volume and green weight associated with that cut tree and adds those yields to running totals for that tree's stand. After the last cut tree is processed, those per acre yields are passed on to the Accountant sub-module.

Accountant takes in the per-acre yields from Tree Merchandiser and uses them to calculate per acre costs as a function of the machine time required to handle the volume removed. It does this by calculating the cost of the harvest system and the cost to load and transport all saw-quality material and biochar feedstock produced by the treatment. Accountant assumes a tether-equipped cut-to-length harvest system will be used for all treatments, with a harvester felling and bucking and a forwarder yarding and loading the material onto trucks for transport. All stands with slope of greater than or equal to 35% are modeled as though all harvester and forwarding operations in the stand are tethered, and all stands with slope of less than 35% are modeled as though all harvester and forwarding operations in the stand are untethered.

Harvester time is calculated as:

(2) $ht = 1.43tw + 0.25hdi$

Where: ht is the harvester time required in productive machine minutes

tw is the total weight of material handled in green tonnes, and

hdi is the distance traveled in meters.

The travel distance, hdi, is held constant at 531 meters per acre and was calculated as two times the linear distance required to cover one acre with a corridor 15 meters wide. This may overestimate the actual distance required on flat ground where opportunistic wandering could allow the harvester to move from section to section without backtracking. It may also underestimate the actual distance on short, steep corridors where it fails to account for travel time between corridors. The harvester time in minutes is then converted to hours and multiplied by either $278.95 per productive machine hour for tethered operations or $212.47 per productive machine hour for untethered operations, producing a harvester cost per hectare.

Forwarding time is calculated as:

(3) ft = 0.05fdi + 2.19sac + 4.06pac on tethered ground, and

(4) ft = 0.01fdi + 2.19sac + 4.06pac on untethered ground

Where ft is the forwarding time required in productive machine minutes

sac is the weight of sawlog material in green tonnes

pac is the weight of feedstock material in green tonnes, and

fdi is the distance traveled by the forwarder in meters.

fdi is calculated by multiplying the number of bunks per acre by the two times the average yarding distance. The number of bunks per acre is calculated by dividing the total green tonnes per acre by twelve and rounding to the nearest whole number. fdi is also subject to a minimum distance of 473 meters per acre, calculated as two times the linear distance required to cover one acre with a corridor 18 meters wide, with an additional thirty meters added per bunk to allow for out-of-corridor travel to and from a pile. The forwarding time in minutes is then converted to hours and multiplied by either $198.11 per productive machine hour for tethered operations or $172.22 per productive machine hour for untethered operations, producing a forwarding cost per hectare.

Truck loading time is calculated as:

(5) lt = (sac / lsp) + (pac / lpp)

Where: lt is the time spent loading in productive machine hours

lsp is the loading rate for saw material in green tonnes per hour, and

lpp is the loading rate for feedstock material in green tonnes per hour.

A per acre loading cost is then calculated as the loading time multiplied by $213.22 ($172.22 per productive machine hour for the forwarder plus $41 per productive forwarder hour for the idle truck).

Haul costs are calculated by identifying the minimum cost route (from the haul cost database) for each piece of saw-quality material and/or biochar feedstock from a given plot to an appropriate facility and summing them for a 0.4 hectare unit (one acre) of each stand. Lastly, Accountant sums the per unit harvester, forwarding, loading, and haul costs into a single per unit treatment cost. This per unit cost does not include fixed per entry costs such as mobilization, which are handled by the Optimizer module.

After all costs, revenues, and yields have been calculated for a prescription, it passes through one final sub-module before the final results are exported to the Optimizer. Fire Resistance Evaluator calculates the Composite Resistance Score (CRS) for each stand under each prescription. This scoring system was designed, in consultation with fire and fuels managers in the western U.S., to generate a metric suitable for evaluating the effectiveness of forest restoration treatments from the perspective of reducing the extent (e.g., percent of crowns burned or scorched) and probability of a stand experiencing crown fire and tree mortality when a fire occurs under fire weather that is sufficiently severe to generate flame lengths of 1.8 to 2.4 meters (Fried et al 2017b). Though designed to address resistance to fire, the score may also serve as a proxy for promoting forest health and resistance to other disturbance and mortality agents such as insects, disease and competition-induced mortality, given that some of its components would bear on those issues as well.

CRS is comprised of four components:

- Canopy base height / crown connectivity
- Canopy bulk density
- Basal area of fire resistant species, and
- Predicted volume mortality.

Each component is assigned a subscore of 0, 1, 2 or 3 based on thresholds from the literature or reflecting practical, though subjective, considerations relating to what resistance means. The four scores are then summed to provide a CRS between zero and twelve, with zero denoting a stand with no resistance and twelve a stand with very high resistance. While the four components are intended to capture different aspects of resistance, they are not entirely independent. For example, a stand with a high proportion of basal area in fire resistance species (component 3) will likely also have lower values of canopy bulk density and higher values of canopy base height. Components 1 and 2 are specifically targeted at resistance through reducing crown fire probability, while components 3 and 4 are focused on fire effects under any kind of fire. As implemented in analyses to date (Fried et al 2017b), any improvement in CRS compared to a status-quo alternative such as no management, is considered effective and desirable, while any reduction in CRS is viewed as counter-productive to achieving fire hazard reduction, and potentially, other restoration objectives.

The canopy base height / crown connectivity is calculated in one of two ways. For stands with a single stratum (as defined by FVS defaults) Fire Resistance Evaluator uses the canopy base height output from the FVS structure class table. For multi-strata stands, Fire Resistance Evaluator uses canopy connectivity, calculated as the distance between the height of the tallest tree of the lowest height stratum (which will have the highest FVS stratum number) and the lowest canopy base height of the second lowest height stratum. Canopy base height and crown connectivity are both scored in the same fashion:

- Zero points for heights / distances of less than 2.1 meters (seven feet)
- One point for heights / distances between 2.1 and 6.1 meters (twenty feet)
- Two points for heights / distances between 6.1 and 9.1 meters (thirty feet), and
- Three points for heights / distances of 9.1 meters or greater.

Canopy bulk density is taken directly from the FVS Potfire tables and is scored as follows:

- Zero points for densities of 0.15 kilograms per cubic meter or greater
- One point for densities between 0.1 and 0.15 kilograms per cubic meter
- Two points for densities between 0.05 and 0.1 kilograms per cubic meter, and
- Three points for densities of 0.05 kilograms per cubic meter or less

The basal area of resistant species is calculated as percent of total (expressed as a fractional value between zero and one). In our study area, the literature suggests six species as comparatively fire resistant: ponderosa, Jeffrey and sugar pine; Douglas-fir; and for larger diameter trees, red and Shasta fir (Agee 1993, Jain et al. 2012). Total basal area and resistant basal area are calculated using FVS tree list records, with the tree list TPA adjusted to reflect TPA removed during treatment. The DBH of each tree is squared and multiplied by 0.00007854 and the tree's frequency to get the basal area per acre for that tree record. That basal area is then added to the running total for the stand and (depending on species) the running total for resistant species in the stand. Resistant basal area fraction (RBA) is scored by assigning:

- Zero points for an RBA of less than 0.25
- One point for an RBA between 0.25 and 0.5
- Two points for an RBA between 0.5 and 0.75, and
- Three points for an RBA of 0.75 or greater.

Predicted volume mortality is the estimated mortality in the event of a fire with 1.8 to 2.4 meter flame lengths, as a percent of the total volume. Predicted volume mortality is calculated using parameters provided by the authors of Fried et al (2017b) by multiplying the volume for each tree by the appropriate predicted mortality parameter (Table 3.8), summing the predicted volume mortality, and dividing it by the total volume. After calculation, the predicted volume mortality is scored with:

- Zero points for predicted volume mortality of greater than 0.98
- One point for predicted volume mortality between 0.98 and 0.7
- Two points for predicted volume mortality between 0.7 and 0.4, and
- Three points for predicted volume mortality of 0.4 or less.

Composite resistance scores are calculated for each of five years of interest, the four treatment years (one, six, eleven, and sixteen) and the last year in the planning horizon (year twenty). In years where treatment occurs, the composite resistance score is always calculated for post-treatment conditions. Once the yields, revenues, costs, and scores are all calculated, the Evaluator module packages them into an ordered list, with one such list for every valid stand/prescription combination, and exports those lists to be used by the Optimizer module.

Table 3.8: Predicted mortality parameters used in calculating Composite Resistance Scores by species and diameter at breast height (DBH) class

DBH (cm)	<12.7	12.7-25.3	25.4-38.0	38.1-53.2	53.3-76.1	76.2-101.5	>=101.6
DBH (inches)	<5	5-9.9	10-14.9	15-20.9	21-29.9	30-39.9	>=40
White fir	0.99	0.99	0.65	0.46	0.14	0.09	0.065
Grand fir	1	0.9675	0.94	0.7325	0.47	0.47	0.47
Subalpine fir	1	0.9675	0.94	0.7325	0.47	0.47	0.47
Incense cedar	0.95	0.95	0.95	0.5	0.11	0.025	0.03
Engelmann spruce	0.99	0.99	0.955	0.73	0.72	0.655	0.655
Red fir	0.85	0.535	0.295	0.1425	0.09	0.09	0.09
Shasta fir	0.85	0.535	0.295	0.1425	0.09	0.09	0.09
Whitebark pine	1	0.99	0.9075	0.775	0.37	0.18	0.18
Lodgepole pine	1	0.99	0.9075	0.775	0.37	0.18	0.18
Sugar pine	1	1	0.905	0.56	0.2	0.16	0.125
Jeffrey pine	0.8	0.8	0.66	0.4475	0.2025	0.0725	0.06
Ponderosa pine	0.8	0.8	0.66	0.4475	0.2025	0.0725	0.06
Douglas-fir	0.98	0.98	0.705	0.54	0.275	0.305	0.215
Quaking aspen	1	1	1	1	1	1	1
Western juniper	1	0.991	0.951	0.839	0.708	0.596	0.47
Rocky mountain maple	1	0.991	0.951	0.839	0.708	0.596	0.47
Chinkapin	1	0.991	0.951	0.839	0.708	0.596	0.47
Mountain mahogany	1	0.991	0.951	0.839	0.708	0.596	0.47
Chokecherry	1	0.991	0.951	0.839	0.708	0.596	0.47
Oregon white oak	1	0.991	0.951	0.839	0.708	0.596	0.47
California black oak	1	0.991	0.951	0.839	0.708	0.596	0.47
Pacific silver fir	0.997	0.991	0.868	0.607	0.332	0.157	0.093
Western white pine	0.997	0.991	0.868	0.607	0.332	0.157	0.093
Mountain hemlock	0.997	0.991	0.868	0.607	0.332	0.157	0.093

Neo-Processor - Optimizer

The Optimizer module is run separately from Evaluator and operates in three phases:

- Initialization
- Optimization, and
- Outputting

Commented [JP3]: Reverted this from "Output" to "Outputting" for the sake of delineating between process and product.

Initialization begins by importing the stand level data from Evaluator. This data is stored internally on a stand by stand basis. Those stands are then sorted into two groups, those eligible for at least one fuels treatment, and those that are ineligible for any treatment. To save on processing time, stands ineligible for fuels treatments within the planning horizon are categorically excluded from optimization. Within the project area 354 of 825 stands (representing 45% of total federal forested hectares) are excluded in this way.

To allow for a more granular solution space, each stand record is broken into smaller, independent fragments for analysis and optimization. During initialization, the area of each stand is assessed one by one. If the area of the stand is less than the fragment size, that stand is used to create a single fragment. If the area is larger than the fragment size, a fragment of that size is created and the area for the stand is reduced by the same amount. This process is iterative, repeating until the number of acres remaining is less than the fragment size, and the remaining area made into one final fragment. To illustrate, for a stand of 3500 hectares and a fragment size of 1000 hectares, that stand would create three fragments of 1000 hectares each, and one fragment of the remaining 500 hectares. Each of those four fragments would be tracked separately and assigned treatments as if they were wholly different operational units. A fragment size of 40.5 hectares (100 acres) was used for all scenarios presented here.

Once all fragments are created and populated with the appropriate yields and CRS values, a partially random initial solution is generated as a two-step process. Fifty percent of fragments to be optimized are assigned a prescription at random from a list of prescriptions valid for that fragment (including a no-action alternative). If an assignment would violate the limitation on area treated in any period, that fragment is assigned the no-action alternative instead. The remaining fragments are used to generate a randomized list for guided prescription assignment. During guided prescription assignment the model:

- Finds the treatment period with the smallest area treated.
- Cycles through the list until it finds a fragment with a treatment capable of increasing the area treated in that period.
- Assigns that fragment a prescription that improves the minimum (selecting a prescription at random if more than one treatment is capable of doing so)
- Updates the area treated, and
- Removes the fragment from the list.

As with random assignment, if guided prescription assignment would violate the limit on area treated, that fragment is ignored as though it had no valid prescriptions capable of improving the minimum. This process repeats until either all fragments have been assigned a treatment, or no fragments are capable of improving the minimum. When no fragments are capable of improving the minimum, all remaining fragments are assigned the no-action alternative. With the outputs for each fragment calculated and the initial solution generated, the model begins the actual work of optimization.

During optimization a random fragment is selected, the model looks up a list of all prescriptions that could be applied to that fragment, and selects a new one at random (excluding the current prescription). The model then checks if the model violates any constraints (described in the scenario section), and rejects the move if it does. Moves that do not violate any constraints are then tested using a modified Great Deluge algorithm (Dueck 1993). Moves that improve the objective function are automatically accepted. The objective function and constraints can be expressed as the following mathematical model:

6) $MAX \sum_i \sum_t \sum_p CRS_{itp} \cdot X_{itp} \cdot A_i \qquad i = 1\ to\ N; \quad p = 1\ to\ P; \quad t \in T^i$

Subject to:

7) $\sum_t \sum_p X_{itp} = 1 \qquad i = 1\ to\ N; \quad p = 1\ to\ P; \quad t \in T^i$

8) $\sum_i \sum_t A_i \cdot X_{itp} \leq TRT_p \qquad i = 1\ to\ N; \quad p = 1\ to\ P; \quad t \in T^i$

9) $0.8 \cdot TRT_{ave} \leq \sum_i \sum_t A_i \cdot X_{itp} \leq 1.2 \cdot TRT_{ave} \qquad i = 1\ to\ N; \quad p = 1\ to\ P; \quad t \in T^i$

10) $\sum_i \sum_t \ CT_{itp} \cdot X_{itp} \leq \sum_i \sum_t \ RT_{itp} \cdot X_{itp} \qquad i = 1\ to\ N; \quad p = 1\ to\ P; \quad t \in T^i$

11) $X_{itp} \in \{0,1\}$

Where:

X_{itp} is a binary variable equaling 1 when fragment *i* is treated with treatment *t* in period *p*

T^{i} is the eligible treatments for group *i*

CRS_{itp} is the composite resistance score of fragment *i*, treatment *t*, period *p*

A_{i} is the area of group *i*

TRT_{p} is the upper limit of area treated in period *p*

TRT_{ave} is the average area treated per period

CT_{itp} is the cost of treatment *t,* fragment *i*, in period *p*, and

RT_{itp} is the revenue from treatment *t*, fragment *i*, period *p*

Equation six maximizes the area-weighted composite resilience scores across the landscape. Equation seven enforces the limit of one treatment per fragment. Equation eight requires that the total area treated in each period be less than or equal to an upper limit on area treated for the period. Equation nine enforces an even flow constraint, requiring the area treated in any period to deviate from the average for all periods by no more than twenty percent. Equation ten requires the revenue generated in each period to be equal to or greater than the cost of implementing all treatments in that period, transporting merchantable material to an appropriate facility, and (in some scenarios) disposal of non-merchantable material. Equation eleven requires that the fragment be treated or assigned to the no-action alternative.

Moves that reduce the objective function may also be accepted as an allowable disimprovement. What constitutes an allowable disimprovement changes over the course of each run according to the following rules:

- An allowable disimprovement may be no worse than the current objective function minus the “flood” value.
- The flood value is set to zero during solution generation and defaults back to zero if it would ever become negative.
- If the flood value is zero and an improvement is found, the flood value is set to 99% of the value of that improvement.
- If the flood value is a non-zero number, an improvement is found, and the improvement is less than or equal to ten times the current flood value, the current flood value is reduced by 50% of the value of the improvement. Lastly,

- If the flood value is not zero, an improvement is found, and that improvement is more than ten times the current flood value, the flood value is set to 99% of the value of that improvement.

This resetting flood system is designed to force the model to become less and less accepting of disimprovement when repeatedly making small gains and disallow backtracking (where the model cycles back and forth between functionally identical moves), but still allow it to effectively explore the solution space when making gains in leaps and bounds.

Optimizer's move evaluation also includes a "noise generation" function, designed to prevent the model from becoming trapped in local maxima for extended periods of time when approaching the limitations on the number of hectares treated. If enough moves are rejected for violating the limitation on hectares treated, noise generation is triggered. When noise generation is triggered, fragments are randomly selected and set to the no-action alternative until 10% of all fragments have been reset in this way.

When a move is accepted, it is immediately implemented and the value of the objective function is recalculated to reflect the change. If the new objective function value is equal to or better than the best objective function value ever seen, that solution is saved away for future use. Optimizer attempts a set number of moves in each run. The total number of moves is determined by the user as a function of the number of fragments being analyzed. For this study, we allowed one thousand moves per fragment, resulting in a total move count of nearly ten million per run. After running through the entire move count for a given run, Optimizer returns the best solution found during that run.

Analytical Scenarios

To explore a range of possible future conditions we prepared thirteen analytical scenarios spanning three levels of treatment capacity and four potential feedstock outcomes. The levels of treatment capacity are:

- 7,689 hectares (19,000 acres) per year, approximating a 100-year management return interval

- 15,387 hectares (38,000 acres) per year, approximating a 50-year management return interval, and
- 30,756 hectares (76,000 acres) per year, approximating a 25-year management return interval.

Each scenario was run thirty times. The average result and the standard deviation of results were calculated to quantify the relative reliability of the model, given the stochastic design elements.

Unconstrained Operation and the No Action Alternative

An unconstrained scenario (WUN) is included to establish the approximate maximum gains that could be made on the landscape given our prescription set. The unconstrained scenario has no limits on the number of acres treated and no even flow or net revenue restrictions. To establish a lower-bound on landscape level composite resistance scores, the results of a no-action alternative (NAA) were calculated deterministically.

Market Scenarios

To determine the effects of biochar price on individual treatment feasibility and treatment allocation, three price points are explored. “No market” scenarios deliver all biochar feedstock material to the Worden facility without generating any revenue. In “partial market” scenarios, the processing facility pays the average haul cost associated with delivering the feedstock from any plot to the facility ($12 per green metric ton), but does not pay for stump to truck costs. In “full market” scenarios, the processing facility pays the average haul cost as described above, and an additional $28 per green metric ton to pay for stump to truck costs (the average estimated cost per ton found in Petitmermet et al. in preparation).

Market scenarios are labeled with NM (no market), PM (partial market), or FM (full market) followed by the approximate management return interval (MRI). For example, PM100 would be the partial market scenario with a 100-year MRI (treating no more than 7,689 hectares annually), while FM25 would be the full market scenario with a 25-year MRI (treating no more than 30,756 hectares annually).

Burn-on-Site Scenarios

Burn-on-site scenarios are included to approximate current operations, in which feedstock sized material is typically piled at the landing and eventually burned. Burning feedstock in this way incurs an additional cost of $13.34 per hectare (P. Cheng, personal communication, 2017). Burn-on-site scenarios are labeled with a "B" followed by the MRI. For example, B50 would be the burn-on-site scenario with a 50-year MRI (treating no more than 15,387 hectares annually).

Subsidy Effects

Fried et al. (2016) noted that industrial forest landowners in California have authorized their foresters to invest as much as $494 per hectare ($200 per acre) to subsidize forest operations that deliver a fuel treatment or fire hazard reduction benefit. Given this, we calculated the potential for subsidies at the $247 per hectare ($100 per acre level) or $494 per hectare level to move stands under a given treatment from a debt-incurring status (net revenue of less than zero dollars per hectare) to self-paying status (net revenue greater than or equal to zero dollars per hectare).

Results

Treatment Eligibility and Effectiveness

Based on this analysis of forest inventory plots, of the 767,000 hectares unreserved, forested acres under federal control in the Upper Klamath Basin:

- 414,000 proved eligible (met all associated requirements as described in the prescription modeling section) for treatment over the 20-year planning horizon
- 231,000 were eligible for one or more thin-from-below treatments

- 87,000 were eligible for one or more q-factor treatments, and
- 125,000 were eligible for a pseudo-clearcut.

Of the hectares eligible for a pseudo-clearcut, approximately 27,000 were also eligible for one or more thin-from-below or q-factor treatment.

Each treatment, on average across all acres in the landscape eligible for that treatment, proved effective at improving composite resistance scores for treated stands relative to the no-action alternative (Table 3.9). The degree to which individual subscores (i.e., canopy base height, canopy bulk density, resistant basal area proportion, and predicted mortality proportion) improved varied significantly by prescription type and treatment timing. In general, the thin-from-below and q-factor treatments achieved the greatest gains by reducing canopy bunk density and ladder fuel connectivity. By contrast, most of the resistance benefits from pseudo-clearcut treatments came about via increases in the resistant basal area proportion and reductions in the predicted volume mortality proportion.

No treatments proved effective for every eligible hectare (Table 10). For each treatment, CRS values failed to improve on up to twelve percent of eligible hectares, and CRS values were decreased (resulting in disimprovement) on up to five percent of eligible hectares. Reductions in CRS values due to treatment were found exclusively among thin from below and pseudo-clearcut treatments.

The Effect of Feedstock Outcomes on Individual Treatment Feasibility

On the 414 thousand hectares that were eligible for one or more fuels treatments, 280 thousand (68% of the eligible hectares) had one or more self-paying treatment when biochar feedstock material was piled and burned at the landing. 263 thousand (63% of the eligible hectares) had one or more self-paying treatment under a no market scenario, increasing to 277 thousand hectares (67% of the eligible hectares) under a partial market scenario and 305 thousand hectares (74% of the eligible hectares) under a full market scenario. Biochar price had the largest effect on the prescriptions that produced a higher proportion of smaller diameter material, such as the less intensive thin from below treatments (which harvest fewer large trees), and prescriptions that produced primarily low-value or non-commercial wood, such as the pseudo-clearcuts (Table 11).

Table 3.9: Average Composite Resistance Score improvement over no action alternative for a single hectare at time of treatment

	Average score improvement				
Treatment[a]	Canopy base height	Canopy bulk density	Resistant basal area	Predicted mortality	Total
1-1	1.15	0.91	0.17	0.30	2.53
1-2	0.96	0.96	0.18	0.29	2.39
1-3	0.91	0.86	0.17	0.24	2.19
1-4	0.83	0.84	0.12	0.19	1.97
2-1	1.04	1.18	0.34	0.43	2.98
2-2	0.85	1.18	0.34	0.39	2.77
2-3	0.86	1.18	0.37	0.39	2.79
2-4	0.71	1.16	0.36	0.39	2.62
3-1	0.87	1.19	0.39	0.38	2.83
3-2	0.78	1.19	0.39	0.35	2.72
3-3	0.72	1.17	0.38	0.37	2.64
3-4	0.62	1.18	0.35	0.36	2.51
4-1	0.66	1.42	0.25	0.25	2.57
4-2	0.87	1.32	0.30	0.23	2.72
4-3	0.86	1.26	0.29	0.35	2.75
4-4	0.88	1.23	0.27	0.29	2.67
5-1	0.70	1.26	0.36	0.24	2.56
5-2	0.91	1.19	0.35	0.26	2.71
5-3	0.91	1.19	0.33	0.28	2.71
5-4	0.99	1.24	0.34	0.27	2.84
6-1	0.34	0.73	2.49	1.24	4.81
6-2	0.63	0.86	2.59	1.42	5.50
6-3	0.65	0.92	2.61	1.51	5.69
6-4	0.65	0.97	2.62	1.55	5.78

[a] Treatments are listed here using a treatment type - period of implementation format. For example: treatment 1-1 represents a type one thin-from-below performed in the first planning period

Table 3.10: Area eligible for treatment and by treatment effect on Composite Resistance Scores (CRS) across the landscape (thousands of hectares)

Treatment	Area eligible for treatment	Area where CRS increased	Area where CRS decreased	Area with no change in CRS
1-1	42.7	40.6	0.0	2.1
1-2	49.0	44.1	0.0	4.9
1-3	51.5	47.2	0.0	4.4
1-4	56.6	49.6	0.0	7.0
2-1	99.8	95.2	0.0	4.5
2-2	114.3	107.0	0.0	7.3
2-3	117.0	109.6	0.0	7.3
2-4	122.8	114.2	1.3	7.3
3-1	196.0	184.2	2.4	9.3
3-2	195.2	184.9	2.4	7.8
3-3	214.6	191.4	3.7	19.5
3-4	230.8	207.1	2.2	21.6
4-1	40.4	39.6	0.0	0.9
4-2	53.0	50.7	0.0	2.4
4-3	62.8	58.7	0.0	4.1
4-4	80.1	70.0	0.0	10.1
5-1	48.6	47.8	0.0	0.9
5-2	64.0	62.2	0.0	1.9
5-3	72.0	69.1	0.0	2.9
5-4	86.5	81.1	0.0	5.4
6-1	72.3	68.7	3.6	0.0
6-2	108.8	105.2	3.7	0.0
6-3	119.5	115.8	0.7	3.0
6-4	124.6	120.1	1.4	3.0

Table 3.11: Total area eligible for treatment and area capable of self-paying (net revenue >= 0 dollars) for each treatment and feedstock outcome (thousands of hectares)

Treatment	Area eligible for treatment	Area capable of self-paying			
		B[a]	NM[b]	PM[c]	FM[d]
1-1	42.7	4.3	3.6	4.3	9.3
1-2	49.0	10.2	6.8	7.2	13.2
1-3	51.5	9.5	6.9	9.5	14.5
1-4	56.6	7.6	6.6	7.6	14.8
2-1	99.8	60.7	55.2	60.7	69.5
2-2	114.3	75.1	70.9	73.5	79.5
2-3	117.0	70.0	66.0	68.9	82.4
2-4	122.8	74.8	69.1	74.0	86.3
3-1	196.0	120.0	112.9	117.7	134.6
3-2	195.2	130.3	117.9	124.6	142.5
3-3	214.6	141.7	136.7	140.9	158.3
3-4	230.8	150.4	140.8	147.7	165.2
4-1	40.4	28.8	28.6	28.8	31.4
4-2	53.0	41.0	40.1	41.0	45.5
4-3	62.8	44.8	43.0	44.1	45.9
4-4	80.1	57.4	54.2	57.4	64.4
5-1	48.6	39.0	38.6	38.7	41.8
5-2	64.0	55.4	53.6	55.4	57.6
5-3	72.0	64.4	62.9	64.4	66.5
5-4	86.5	73.6	69.0	73.6	76.8
6-1	72.3	22.3	15.6	20.8	30.2
6-2	108.8	26.3	22.2	25.6	39.8
6-3	119.5	36.3	28.8	34.3	45.5
6-4	124.6	44.1	37.1	43.3	57.0

[a] Feedstock is burned at the landing at a cost of $13.34 per hectare
[b] Feedstock is delivered to Worden processing facility without generating revenue
[c] Feedstock is delivered to Worden processing facility generating $12 per green metric ton in revenue
[d] Feedstock is delivered to Worden processing facility generating $40 per green metric ton in revenue

The Potential for Subsidies to Improve Individual Treatment Feasibility

Both the $247 and $494 per hectare subsidies proved effective at moving hectares with negative net revenue to a self-paying status. On average, a $247 per hectare subsidy improved three to four percent of eligible hectares with a negative net revenue to self-pay status (Table 3.12). On average, the $494 per hectare subsidy had a similar effect, improving six to ten percent of eligible hectares with a negative net revenue to self-pay status (Table 3.13).

Scenario Results - Acres Treated

All scenarios with 100-year and 50-year MRI (management return interval) treatment capacities were strictly limited by those capacities (Table 3.14). In all eight scenarios with those constraints the average number of acres treated came to within five acres of the annual limit. The scenarios with a 25-year MRI treatment capacity treated an average of 19,800 hectares annually (64% of the maximum allowed). The unconstrained action scenario consistently treated more than half the eligible area in the first period, then treated new stands in the periods in which they grew into eligibility until approximately 408 thousand hectares were treated (of the 414 thousand eligible for treatment at any point in the planning horizon).

Scenario Results - Improvements to Composite Resistance Scores

The no-action alternative found a sum of CRS values of 34.0 out of a potential sixty (five time points with a maximum possible CRS of twelve at each point). The unconstrained action scenario (WUN) found a CRS sum of 50.1, establishing an effective maximum for the landscape with this treatment set. All scenarios with a given capacity constraint found functionally identical gains in composite resistance scores (Table 3.15). Scenarios with a 100-year MRI found sum of CRS values of 38.7, 13% higher than the no-action alternative and 77% of the effective maximum. Scenarios with a 50-year MRI found sum of CRS values of 43.9, a 29% improvement over the no-action alternative and equivalent to 87% of the effective maximum. Scenarios with a 25-year MRI found sum of CRS values of 45.9, 35% higher than the no action alternative and equivalent to 91% of the effective maximum.

All runs for a given scenario tended to find solutions with near identical CRS values, suggesting a high level of reliability for the Neo-Processor model. Standard deviations for the CRS at any given year of interest ranged from 0.02 for year one under the 100-year MRI

Table 3.12: Total area eligible for treatment and area improved to self-paying status (net revenue >= 0 dollars) by a subsidy of $247 per hectare ($100 per acre), for each treatment and feedstock outcome (thousands of hectares)

Treatment	Area eligible for treatment	Area improved to self-paying status[a]			
		B[b]	NM[c]	PM[d]	FM[e]
1-1	42.7	1.7	0.0	1.7	2.6
1-2	49.0	0.0	0.4	3.1	3.1
1-3	51.5	1.0	2.6	0.0	4.0
1-4	56.6	0.2	1.1	0.2	2.2
2-1	99.8	3.0	4.7	2.3	1.1
2-2	114.3	3.2	3.3	4.3	5.5
2-3	117.0	1.2	3.6	2.3	5.6
2-4	122.8	6.4	1.3	5.5	8.5
3-1	196.0	8.4	7.3	10.6	12.7
3-2	195.2	6.5	11.7	11.5	3.2
3-3	214.6	9.4	7.0	8.1	8.7
3-4	230.8	6.5	7.0	8.6	7.0
4-1	40.4	2.5	0.2	2.4	3.4
4-2	53.0	2.1	0.9	1.6	0.0
4-3	62.8	0.8	1.8	1.6	4.1
4-4	80.1	3.7	5.5	2.6	3.8
5-1	48.6	1.4	1.0	1.6	1.0
5-2	64.0	0.0	1.8	0.0	1.0
5-3	72.0	0.9	2.0	0.9	2.4
5-4	86.5	0.3	1.7	0.3	1.3
6-1	72.3	3.2	4.8	2.2	3.5
6-2	108.8	9.4	3.1	6.4	1.6
6-3	119.5	5.7	4.0	5.5	7.6
6-4	124.6	6.3	4.1	4.0	4.7

[a] The area capable of self-paying without subsidy can be found in table 3.11
[b] Feedstock is burned at the landing at a cost of $13.34 per hectare
[c] Feedstock is delivered to Worden processing facility without generating revenue
[d] Feedstock is delivered to Worden processing facility generating $12 per green metric ton in revenue
[e] Feedstock is delivered to Worden processing facility generating $40 per green metric ton in revenue

Table 3.13: Total area eligible for treatment and area improved to self-paying status (net revenue >= 0 dollars) by a subsidy of $494 per hectare ($200 per acre), for each treatment and feedstock outcome (thousands of hectares)

	Area eligible	Area improved to self-paying status[a]			
Treatment	for treatment	B[b]	NM[c]	PM[d]	FM[e]
1-1	42.7	7.3	2.4	4.3	10.1
1-2	49.0	3.7	3.4	6.6	8.1
1-3	51.5	2.6	2.6	1.0	7.1
1-4	56.6	5.3	1.3	2.9	9.5
2-1	99.8	6.1	8.5	5.3	6.6
2-2	114.3	3.7	6.2	5.3	12.5
2-3	117.0	7.8	5.2	6.7	13.2
2-4	122.8	17.1	12.8	14.6	12.5
3-1	196.0	24.6	18.3	23.2	14.9
3-2	195.2	10.5	17.1	14.1	11.7
3-3	214.6	15.7	11.5	15.6	16.3
3-4	230.8	16.0	18.2	17.1	14.4
4-1	40.4	3.0	2.9	2.8	3.7
4-2	53.0	2.1	2.7	2.1	2.6
4-3	62.8	4.1	5.7	4.7	8.1
4-4	80.1	7.1	6.2	7.1	6.4
5-1	48.6	1.4	1.3	1.6	1.2
5-2	64.0	0.0	1.8	0.0	3.2
5-3	72.0	2.2	2.4	1.2	2.8
5-4	86.5	2.9	3.3	2.1	2.5
6-1	72.3	8.7	6.7	6.5	12.8
6-2	108.8	13.7	7.9	10.0	10.5
6-3	119.5	10.2	9.6	9.9	12.5
6-4	124.6	7.1	10.0	7.0	13.8

[a] The area capable of self-paying without subsidy can be found in table 3.11

[b] Feedstock is burned at the landing at a cost of $13.34 per hectare

[c] Feedstock is delivered to Worden processing facility without generating revenue

[d] Feedstock is delivered to Worden processing facility generating $12 per green ton in revenue

[e] Feedstock is delivered to Worden processing facility generating $40 per green ton in revenue

Table 3.14: Mean annual treatment area (hectares), by 5-year period, over thirty simulations

Scenario	Period 1	Period 2	Period 3	Period 4
B100	7,688	7,688	7,688	7,688
NM100	7,688	7,688	7,688	7,688
PM100	7,688	7,688	7,688	7,688
FM100	7,688	7,689	7,688	7,688
B50	15,376	15,377	15,376	15,376
NM50	15,376	15,377	15,377	15,377
PM50	15,377	15,377	15,377	15,377
FM50	15,376	15,377	15,377	15,376
B25	19,297	21,998	19,120	18,926
NM25	19,335	21,915	19,156	18,880
PM25	19,348	21,895	19,186	18,892
FM25	19,284	22,073	19,100	18,846
WUN	50,202	15,567	9,627	6,225

Table 3.15: Mean Composite Resistance Score (CRS), by year, and sum of scores for these five representative years, over thirty simulations for each scenario and the no-action alternative (NAA)

Scenario	Year 1	Year 6	Year 11	Year 16	Year 20	Sum of scores
NAA	6.69	6.78	6.81	6.87	6.91	34.05
B100	6.97	7.41	7.81	8.25	8.29	38.74
NM100	6.97	7.41	7.81	8.25	8.29	38.74
PM100	6.97	7.41	7.82	8.26	8.29	38.74
FM100	6.97	7.40	7.81	8.25	8.29	38.73
B50	7.24	8.13	8.97	9.77	9.80	43.90
NM50	7.24	8.14	8.98	9.78	9.80	43.93
PM50	7.24	8.13	8.97	9.78	9.80	43.92
FM50	7.24	8.13	8.97	9.78	9.80	43.92
B25	7.39	8.62	9.46	10.20	10.22	45.90
NM25	7.40	8.62	9.46	10.20	10.22	45.90
PM25	7.40	8.61	9.46	10.20	10.22	45.89
FM25	7.40	8.62	9.46	10.20	10.22	45.90
WUN	8.98	9.87	10.26	10.50	10.52	50.13

scenarios (0.2% of the associated average score), to 0.08 for year one under 25-year MRI scenarios (1.1% of the associated average score). In general, standard deviations increased slightly as treatment capacity increased.

Scenario Results - Net Revenue and Saw Log Production

Net revenue was positive for all market and burn-on-site scenarios (Table 3.16). Net revenue was sensitive to both treatment capacity and feedstock outcome, with net revenue increasing as treatment capacity increased and as biochar feedstock price increased. The burn-on-site scenario produced net revenues two to five percent greater than partial market scenarios with the same treatment capacity and fifteen to thirty percent less than full market scenarios with the same treatment capacity.

Table 3.16: Mean annual net revenue (millions of dollars) by scenario and five-year planning period over thirty simulations

Scenario	Period 1	Period 2	Period 3	Period 4
B100	29.48	21.91	18.94	18.26
NM100	27.87	20.06	15.97	15.39
PM100	29.57	21.44	18.18	17.44
FM100	33.60	25.69	22.55	22.28
B50	37.24	26.32	24.41	26.49
NM50	31.79	20.53	18.93	20.47
PM50	35.70	24.42	23.46	24.56
FM50	45.65	34.06	32.15	35.01
B25	44.31	25.07	30.64	34.16
NM25	36.97	17.45	24.96	27.62
PM25	41.85	22.71	29.58	32.87
FM25	54.04	35.31	39.59	43.64
WUN	83.55	2.34	-0.54	3.39

The unconstrained action scenario generated a positive net revenue over the whole of the planning horizon, but generated relatively low revenue in periods two and four and ran a deficit in period three. This outcome is driven by the objective function (maximizing the sum of CRS values over the planning horizon) driving the model toward using the earliest, most intensive treatments in as many cases as possible, such that in later periods the only stands available to be treated are those growing into treatment eligibility in that period.

Saw log production, as with CRS, and hectares treated, proved nearly identical for all scenarios sharing the same treatment capacity constraint (Table 3.17), generating an average of approximately 276,000 bone dry tonnes annually for the 100-year MRI scenarios, 446,000 bone dry tonnes annually for the 50-year MRI scenarios, and 545,000 bone dry tonnes annually for the 25-year MRI scenarios. The unconstrained scenario generated an average of 524,000 bone dry tonnes annually, but for the same reasons described above for net revenue, most material produced (72%) was delivered in period one.

Table 3.17: Mean annual saw log production (thousands of bone dry tonnes) by scenario and five-year planning period over thirty simulations

Scenario	Period 1	Period 2	Period 3	Period 4
B100	300	271	255	273
NM100	303	272	253	274
PM100	302	271	255	274
FM100	304	271	257	271
B50	476	414	423	466
NM50	479	417	422	467
PM50	479	416	425	464
FM50	480	418	421	466
B25	569	512	526	574
NM25	568	510	527	577
PM25	567	507	530	579
FM25	566	510	528	577
WUN	1516	257	178	146

Scenario Results - Biochar Feedstock Generation and Average and Marginal Costs

As with previous outputs, biochar feedstock generation varied significantly by treatment capacity, but not by feedstock outcome (Table 3.18). On average, the 100-year MRI scenarios generated approximately 91,000 bone dry tonnes of feedstock material annually, with the 50-year MRI scenarios generating 201,000 bone dry tonnes annually and the 25-year MRI scenarios generating 241,000 bone dry tonnes annually. The unconstrained scenario produced an average of 242,000 bone dry tonnes annually, but as with all other measures, this output was skewed heavily towards the first period, which accounted for 72% of the total delivered.

Table 3.18: Mean annual biochar feedstock production (thousands of bone dry tonnes) by scenario and five-year planning period over thirty simulations

Scenario	Period 1	Period 2	Period 3	Period 4
B100	79	88	90	107
NM100	79	87	88	107
PM100	78	88	90	107
FM100	79	88	89	107
B50	203	195	193	215
NM50	203	196	192	215
PM50	204	195	191	216
FM50	202	196	192	215
B25	257	264	208	237
NM25	258	261	209	238
PM25	259	261	207	238
FM25	257	262	208	238
WUN	704	147	71	47

Two sets of supply curves were prepared for each market scenario and the unconstrained action scenario. “FLH” curves (Figures 3.1 - 3.10) describe the marginal cost per bone dry metric ton of feedstock produced by the average run for that scenario. Costs on the FLH curves include the haul cost, truck loading cost, and forwarding cost, and represent our best estimate for the “true” variable cost of the feedstock material. Costs on the “LH” curves (Figures 3.11 - 3.20) include only the haul cost and truck loading cost. Given that the alternative for feedstock

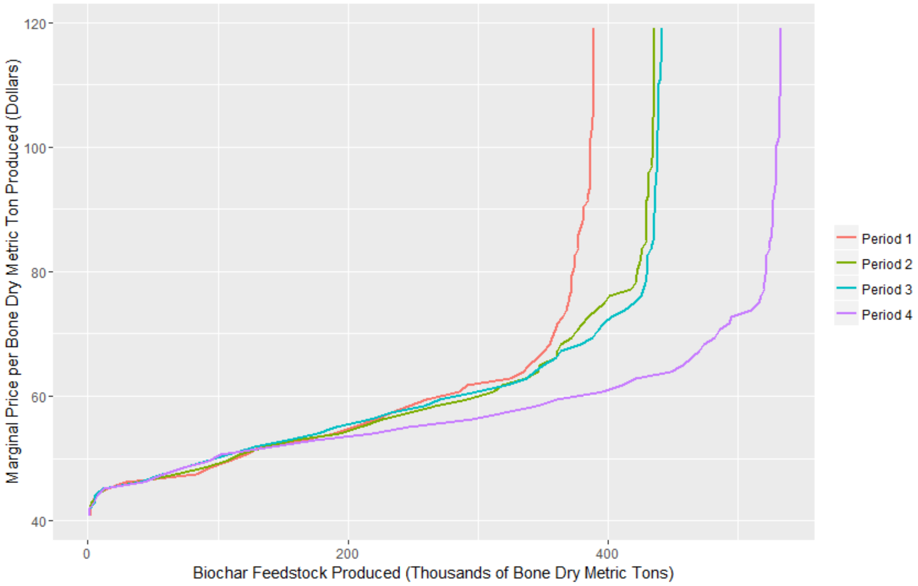


Figure 3.1: Biochar feedstock produced and associated marginal supply costs (including forwarding, loading, and haul) for each five-year planning period of the average solution for the “no market” scenario with a 100-year approximate management return interval

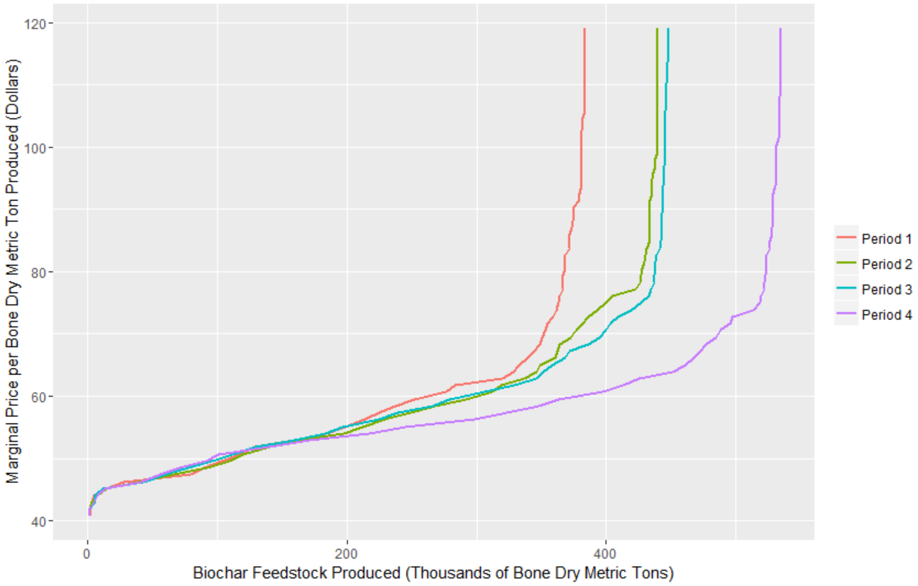


Figure 3.2: Biochar feedstock produced and associated marginal supply costs (including forwarding, loading, and haul) for each five-year planning period of the average solution for the “partial market” scenario with a 100-year approximate management return interval

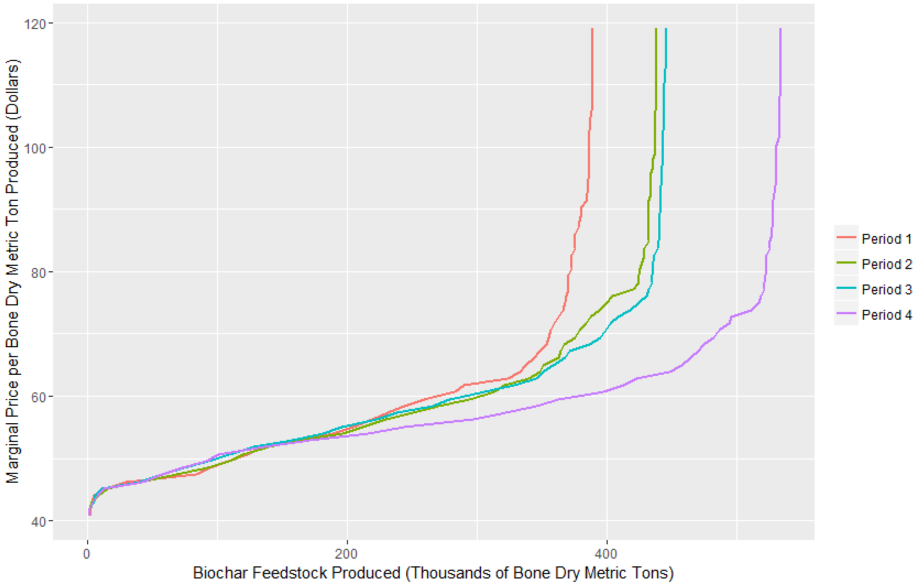


Figure 3.3: Biochar feedstock produced and associated marginal supply costs (including forwarding, loading, and haul) for each five-year planning period of the average solution for the “full market” scenario with a 100-year approximate management return interval

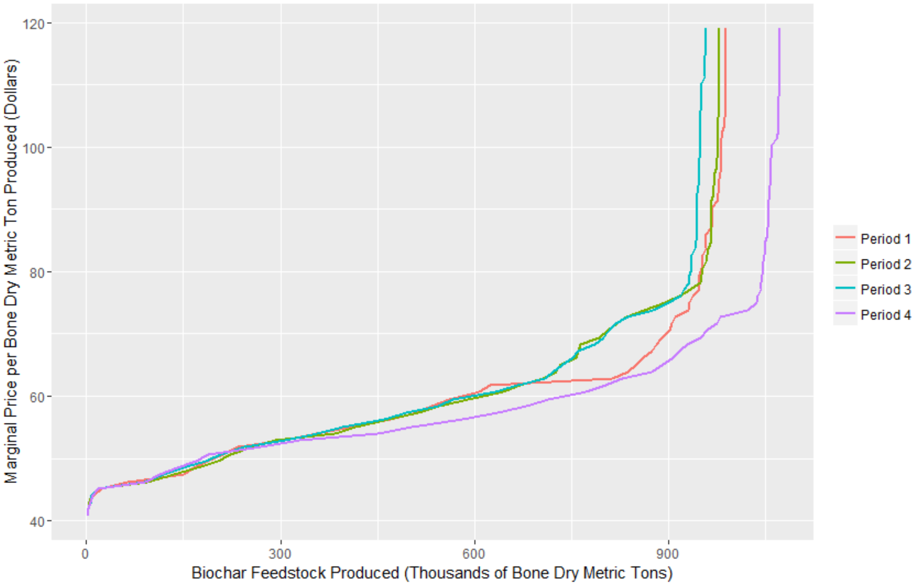


Figure 3.4: Biochar feedstock produced and associated marginal supply costs (including forwarding, loading, and haul) for each five-year planning period of the average solution for the “no market” scenario with a 50-year approximate management return interval

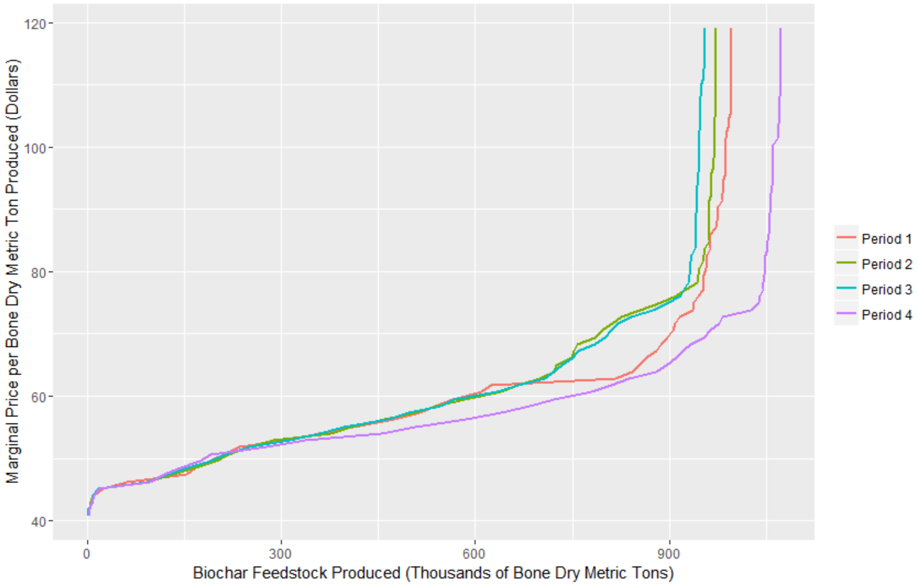


Figure 3.5: Biochar feedstock produced and associated marginal supply costs (including forwarding, loading, and haul) for each five-year planning period of the average solution for the “partial market” scenario with a 50-year approximate management return interval

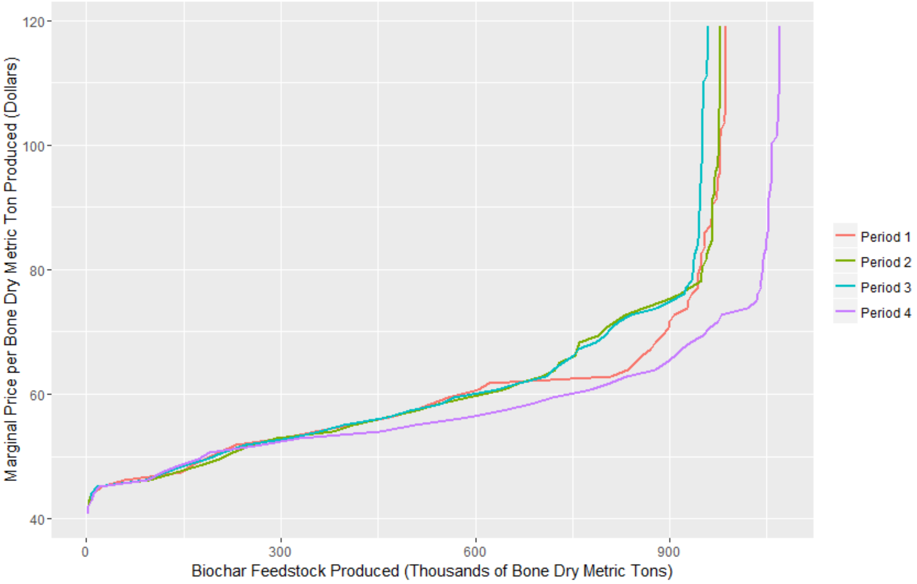


Figure 3.6: Biochar feedstock produced and associated marginal supply costs (including forwarding, loading, and haul) for each five-year planning period of the average solution for the "full market" scenario with a 50-year approximate management return interval

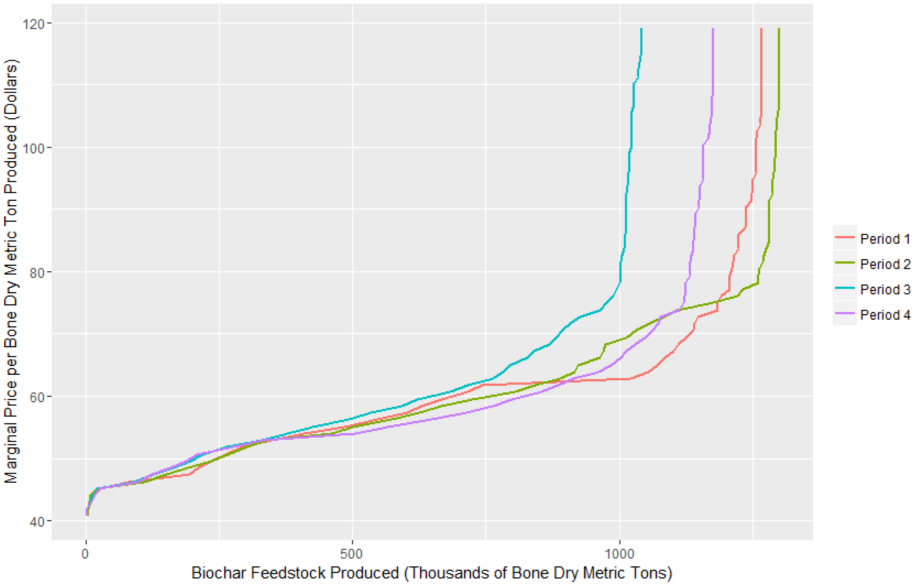


Figure 3.7: Biochar feedstock produced and associated marginal supply costs (including forwarding, loading, and haul) for each five-year planning period of the average solution for the “no market” scenario with a 25-year approximate management return interval

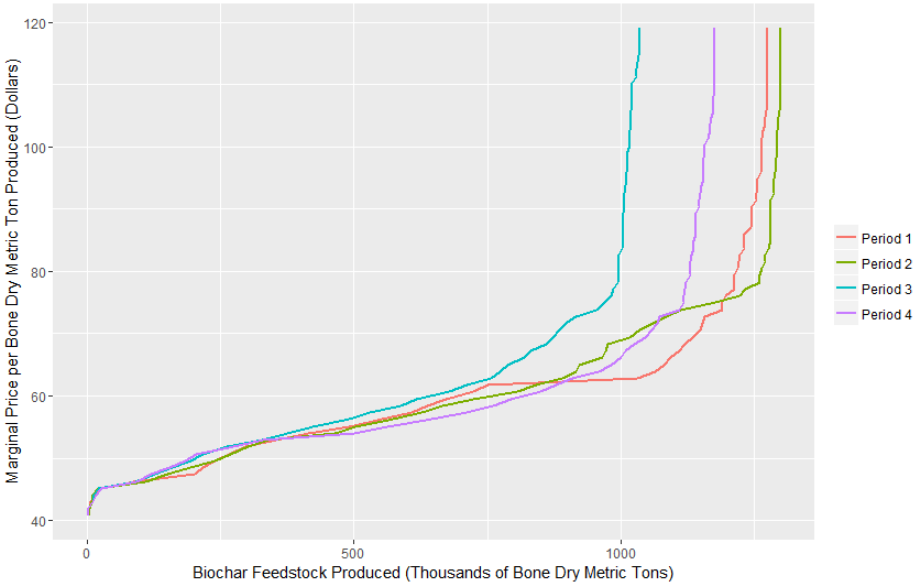


Figure 3.8: Biochar feedstock produced and associated marginal supply costs (including forwarding, loading, and haul) for each five-year planning period of the average solution for the “partial market” scenario with a 25-year approximate management return interval

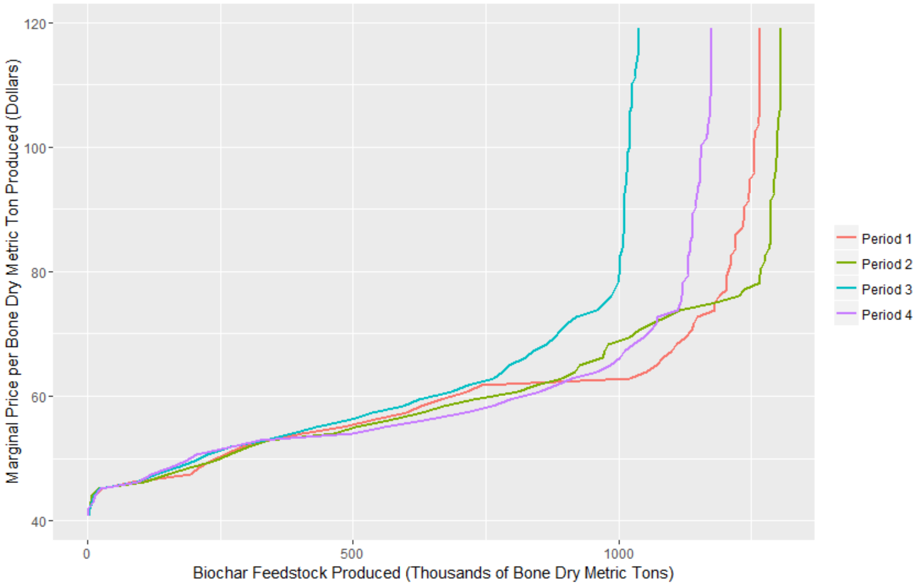


Figure 3.9: Biochar feedstock produced and associated marginal supply costs (including forwarding, loading, and haul) for each five-year planning period of the average solution for the “full market” scenario with a 25-year approximate management return interval

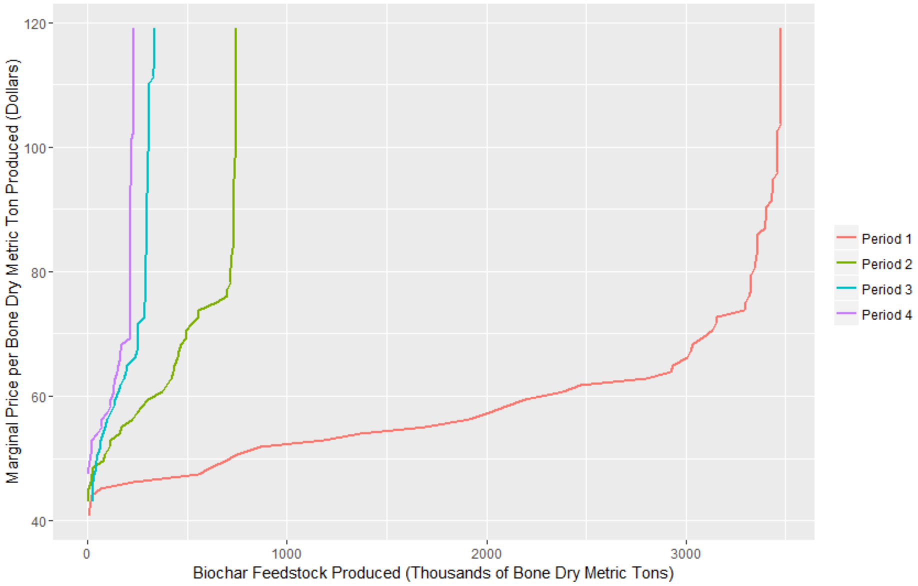


Figure 3.10: Biochar feedstock produced and associated marginal supply costs (including forwarding, loading, and haul) for each five-year planning period of the average solution for the “unconstrained action” scenario

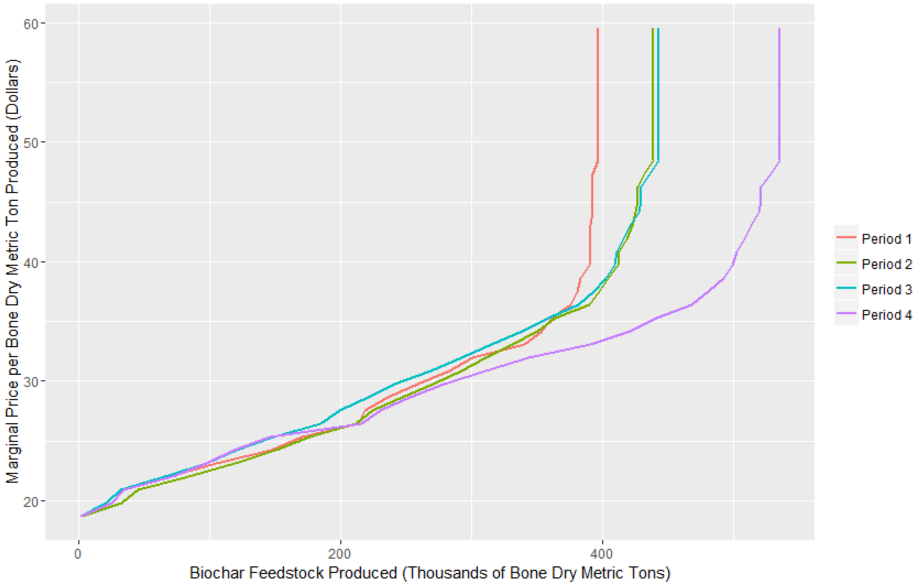


Figure 3.11: Biochar feedstock produced and associated marginal supply costs (including loading and haul only) for each five-year planning period of the average solution for the “no market” scenario with a 100-year approximate management return interval

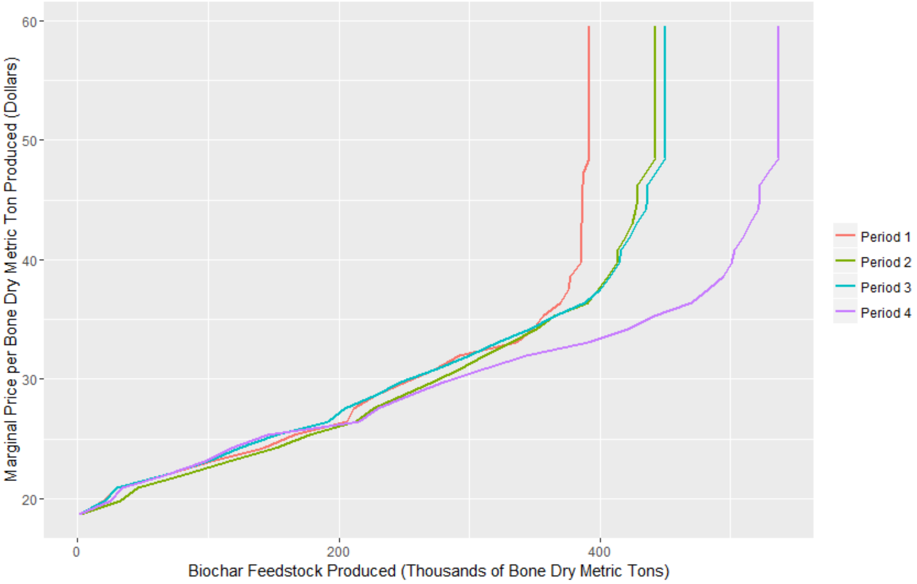


Figure 3.12: Biochar feedstock produced and associated marginal supply costs (including loading and haul only) for each five-year planning period of the average solution for the "partial market" scenario with a 100-year approximate management return interval

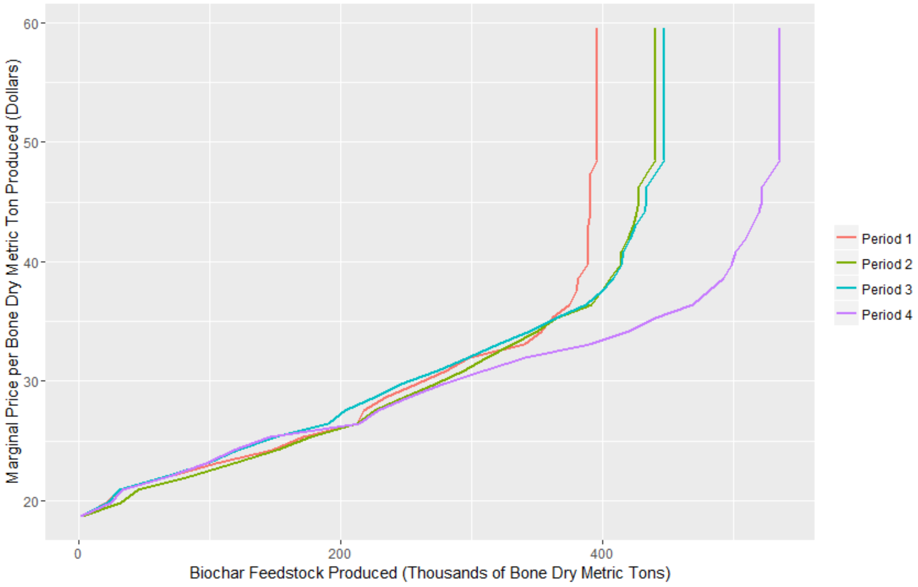


Figure 3.13: Biochar feedstock produced and associated marginal supply costs (including loading and haul only) for each five-year planning period of the average solution for the “full market” scenario with a 100-year approximate management return interval

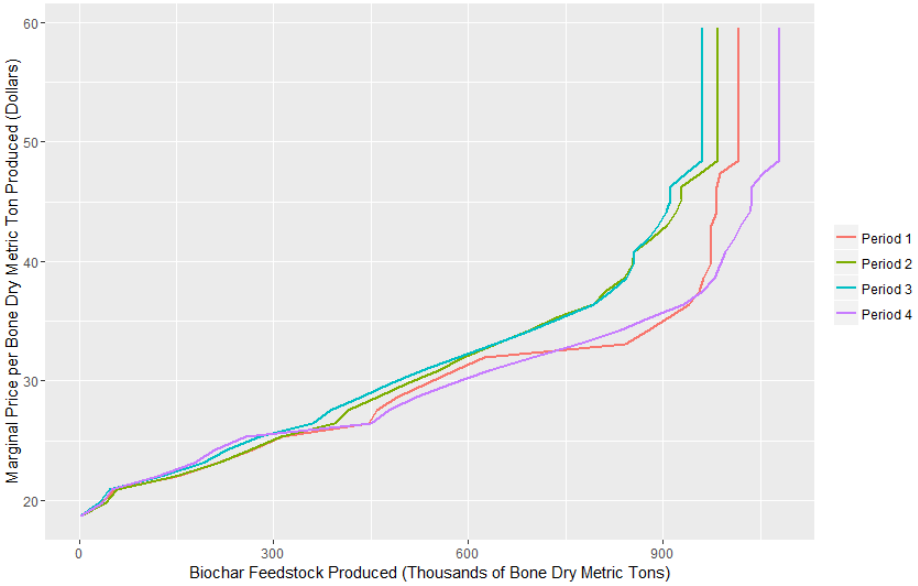


Figure 3.14: Biochar feedstock produced and associated marginal supply costs (including loading and haul only) for each five-year planning period of the average solution for the “no market” scenario with a 50-year approximate management return interval

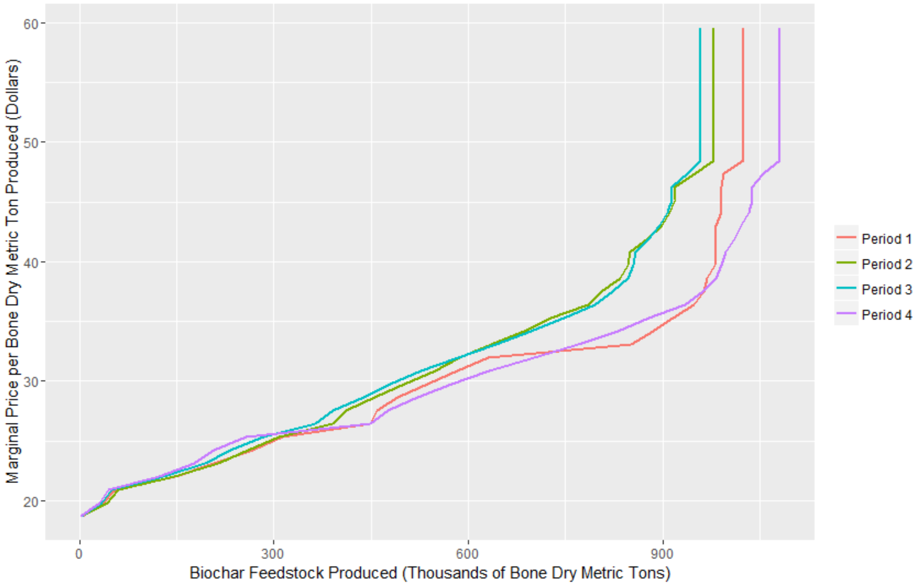


Figure 3.15: Biochar feedstock produced and associated marginal supply costs (including loading and haul only) for each five-year planning period of the average solution for the "partial market" scenario with a 50-year approximate management return interval

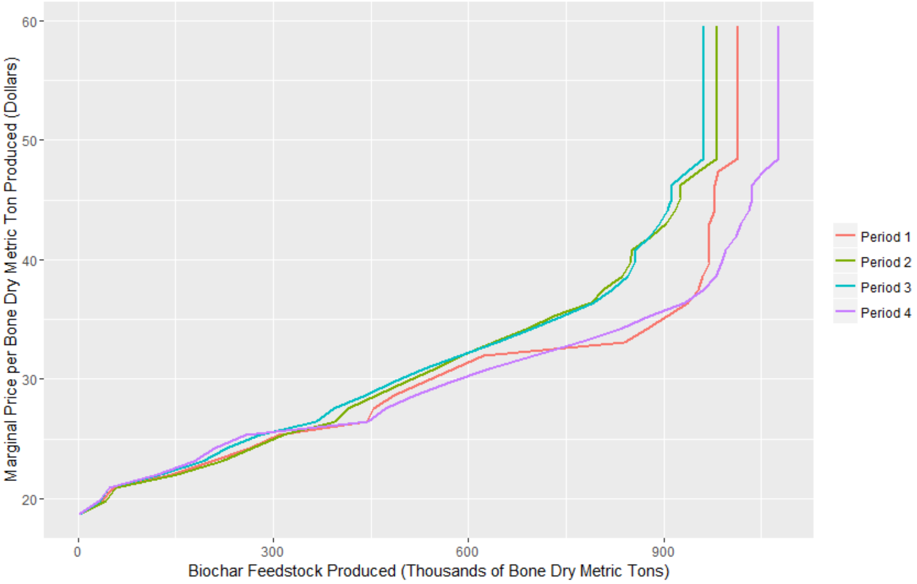


Figure 3.16: Biochar feedstock produced and associated marginal supply costs (including loading and haul only) for each five-year planning period of the average solution for the "full market" scenario with a 50-year approximate management return interval

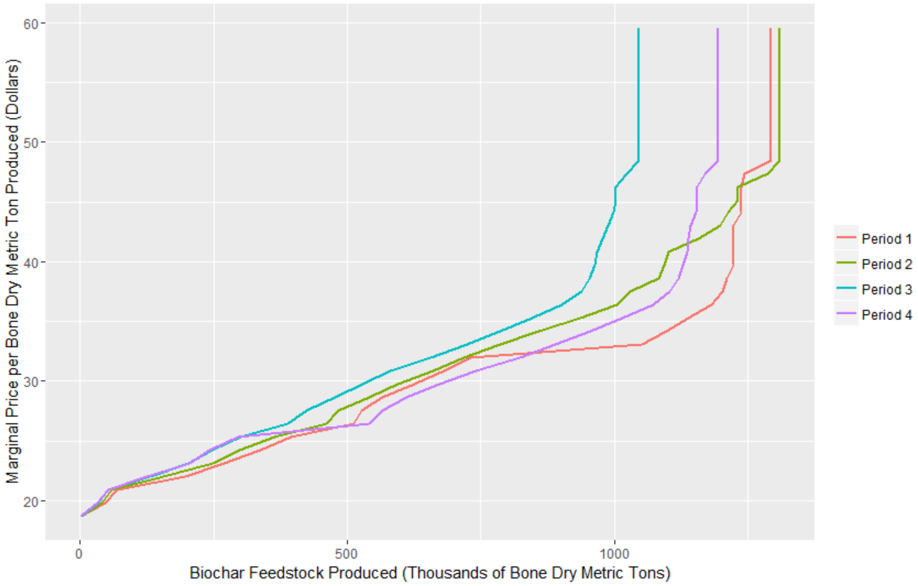


Figure 3.17: Biochar feedstock produced and associated marginal supply costs (including loading and haul only) for each five-year planning period of the average solution for the “no market” scenario with a 25-year approximate management return interval

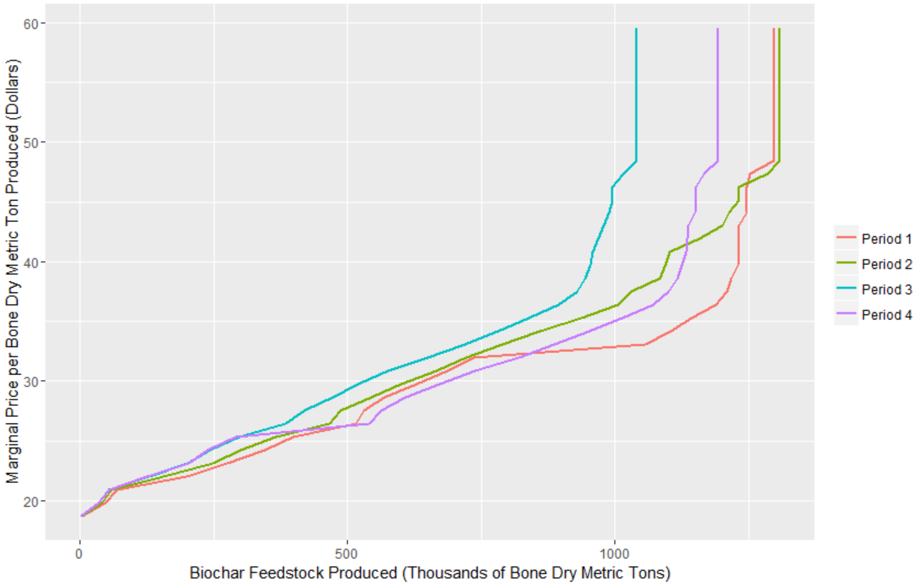


Figure 3.18: Biochar feedstock produced and associated marginal supply costs (including loading and haul only) for each five-year planning period of the average solution for the "partial market" scenario with a 25-year approximate management return interval

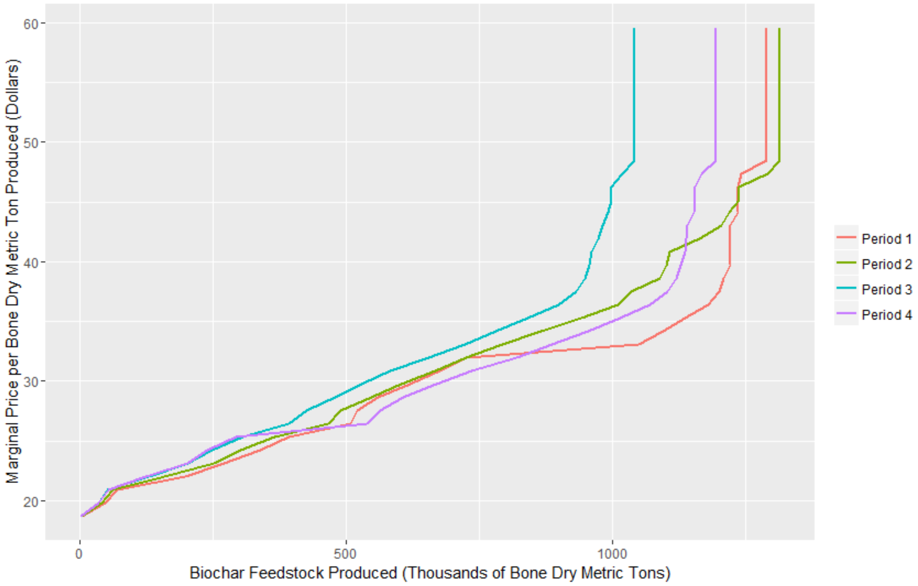


Figure 3.19: Biochar feedstock produced and associated marginal supply costs (including loading and haul only) for each five-year planning period of the average solution for the “full market” scenario with a 25-year approximate management return interval

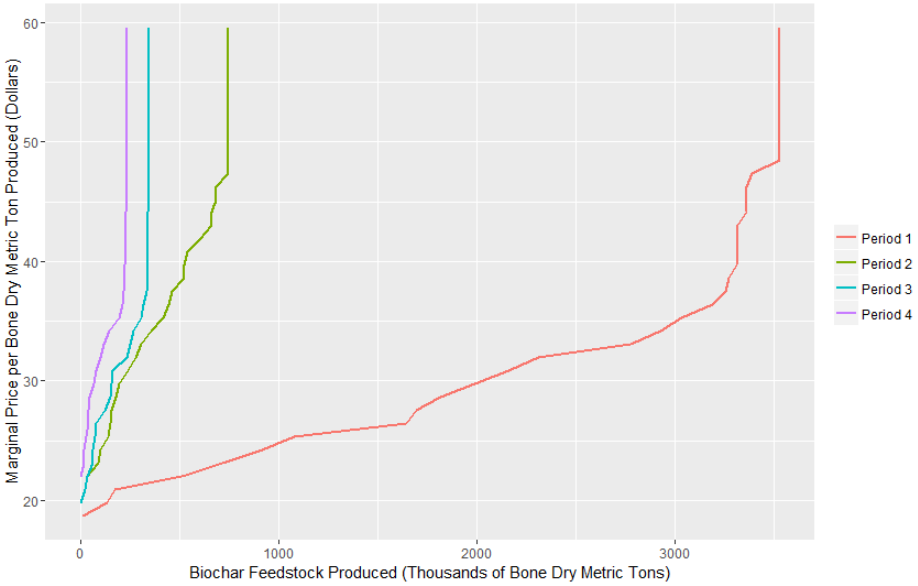


Figure 3.20: Biochar feedstock produced and associated marginal supply costs (including loading and haul only) for each five-year planning period of the average solution for the “unconstrained action” scenario

disposal in the analysis also requires forwarding the material to the landing, the LH curves provide an estimate of the breakeven point where the cost of sending the feedstock to the processing facility is economically equivalent to the cost of burning it on site. As with most of the other metrics, marginal and average costs varied only minimally for different feedstock outcomes with the same treatment capacity constraint.

The marginal cost for the first 227,000 bone dry tonnes per period (45,360 bone dry tonnes annually) ranged from $49 to $56 per bone dry metric ton on the FLH curves and $22 to $27 per bone dry metric ton on the LH curves. In both cases the marginal cost increased as treatment capacity decreased. This increase in cost is driven by the objective function leading the model to preferentially select actions with higher costs and a greater effect on CRS values over stands and/or treatments that could produce feedstock at a lower cost.

The average cost for the first 227,000 bone dry tonnes per period ranged from $39 to $42 per bone dry ton on the FLH curves and $18 to $20 on the LH curves (Table 3.19). As with the marginal costs, the average cost per dry ton increased as treatment capacity decreased.

Table 3.19: Average and costs for the first 250,000 bone dry tonnes of biochar feedstock delivered to the Worden facility per five-year planning period for the FLH[a] and LH[a] curves for each scenario (dollars per bone dry metric ton)

	LH Average				FLH Average			
Period	1	2	3	4	1	2	3	4
NM100	24.10	24.00	24.79	24.17	50.82	51.23	51.35	50.96
PM100	24.09	24.03	24.35	24.18	51.22	50.61	51.31	50.99
FM100	24.07	24.02	24.79	24.18	50.79	50.61	51.39	50.99
NM50	22.36	22.31	22.33	22.80	47.94	48.11	48.25	48.79
PM50	22.40	22.29	22.31	22.83	47.95	48.11	48.23	48.78
FM50	22.41	22.29	22.31	22.82	47.98	48.06	48.25	48.73
NM25	21.85	21.98	22.35	22.35	47.10	47.31	48.30	48.52
PM25	21.86	21.99	22.33	22.31	47.06	47.33	48.31	48.57
FM25	21.85	21.99	22.38	22.33	47.06	47.33	48.30	48.53
WUN	21.35	26.00	27.55	32.29	45.69	52.74	57.19	62.98

[a] FLH curves include the cost of haul, loading, and forwarding while LH curves include haul and loading costs only

Discussion

The results indicate that, even at the lowest estimated capacity limits, the federal forests of the Upper Klamath Basin would be capable of providing more than the 45,360 BDT per year initially sought for biochar production. Our best approximation of current operations suggests the possibility of roughly double that amount, with a clear potential for further increases in feedstock generation tied to increases in treatment capacity. Additionally, biochar feedstock material generated could be delivered to the facility without accompanying revenue and still result in a positive net revenue in more than half of all treatment / stand combinations. It is extremely unlikely that managers would choose to do this while burning remains both socially feasible and economically preferable, but if concerns over air quality, carbon release, and/or the risk of fire escape rise, it could make shipping feedstock material to a processing facility more attractive despite the economic disincentives of haul costs and a zero price.

Contrary to our initial assumptions, revenue did not prove to be limiting or close to limiting in any scenario analyzed. However, this finding does come with a handful of important caveats. First, all estimates of harvest system cost utilized tether-equipped cut-to-length technology. It is possible that the local logging infrastructure is not capable of fielding enough tethered systems to perform the desired treatments, requiring the use of other ground and/or cable-based systems with higher harvest costs. Second, revenues may be highly dependent on the social feasibility of higher intensity treatments. The type one (thin from below) treatment with a max take diameter of 25.4 centimeters (ten inches) was only capable of paying for itself, at most, a third of the time. If the treatment list was more restricted in terms of either max take diameter or the quantity of material removed, it could severely hamper revenue generation. Third, our cost model does not explicitly account for trees incapable of producing at least one eight-foot log of sawlog or biochar feedstock material. This may lead the model to underestimate costs where very small stems (less than 15 cm DBH) make up much of the material to be cut. Lastly, it is possible that a second entry may be required following cut-to-length treatments to reduce surface fuel loading. Cut-to-length systems generally operate on “brush mats”, branches and other material too small in diameter to be sold to local markets which are collected, piled into trails, and driven over to reduce soil impacts. This process rearranges and often masticates the material to some extent, but does not reduce the total fuel loading (mass per unit area). For situations where managers decide that a burn is required to reduce the fuels left in these brush mats, it would likely dramatically increase the total cost of treatment.

Our initial estimate of a $40 stump to facility cost per green metric ton ($69 per BDT), based on the observed costs of a federal fuels treatment study done near the Upper Klamath Basin (Petitmermet et al. in preparation), proved to be a considerable overestimate for the wood basket. Stump to facility costs averaged 32% to 38% lower than that initial estimate, with average prices decreasing as treatment capacity increased. This decrease in price is driven by the objective function maximizing CRS without regard for any other potential objectives, leading the model to favor high-cost stands with a large impact on CRS over stands with lower cost and lower impact. As the number of hectares treated increases, more low-cost low-impact stands can be included.

The findings presented here show promise for the near-term viability of a biochar processing facility, but further research is required before long-term sustainability can be assessed. First, the preferred method of sustainable fuels treatments must be ascertained. This could take the shape of periodic mechanical entry at intervals similar to the historical mean fire return interval, periodic entry at intervals significantly longer than the historical mean fire return interval to reduce cumulative impacts, or even a single “corrective” entry followed by frequent prescribed fire. Each of those three scenarios could plausibly prove ideal for reducing fire hazard while meeting other social, economic, and ecological objectives, and each of those three scenarios has very different implications for an industry that seeks to utilize the small diameter material produced by mechanical treatments.

The second aspect that requires consideration is the role of private lands in providing supply. As previously noted, Fried et al. 2016 found that private landowners in California were often willing to treat acres at a modest loss to improve fire resistance and resilience. Such landowners would likely welcome the opportunity to sell the material produced by their own fuels reduction efforts to any market that will accept them, including a biochar facility. The size, frequency, and cost of these contributions must be understood for a complete picture of the biochar feedstock supply can develop.

Acknowledgements

This study was funded by the Institute for Working Forest Landscapes as a part of the "Opportunities for biochar production to reduce forest wildfire hazard, sequester carbon, and increase agricultural productivity of dryland soils" project grant. The authors would like to thank Paul Cheng of the Fremont-Winema National Forest for his time and support, Sara Loreno of Ecotrust for her aid in writing and executing KCP files, and John Bailey and Jerry Mohr of Oregon State University for aid in prescription generation and technical support respectively.

References

Barrett, S., Havlina, D., Jones, J., Hann, W.J., Frame, C., Hamilton, D., Schon, K., DeMeo, T., Hutter, L. and Menakis, J., 2010. Interagency Fire Regime Condition Class (FRCC) Guidebook, version 3.0. USDA Forest Service, US Department of the Interior, and The Nature Conservancy. http://www. frcc. gov.

Bell, J.F. and Dilworth, J.R., 1988. Log scaling and timber cruising. OSU Book Stores. Inc., Corvallis, Oregon.

Brunson, M.W. and Shindler, B.A., 2004. Geographic variation in social acceptability of wildland fuels management in the western United States. Society and Natural Resources, 17(8), pp.661-678.

Denardo, Eric V. *Dynamic programming: models and applications*. Courier Corporation, 2012.

Dueck, Gunter. "New optimization heuristics: The great deluge algorithm and the record-to-record travel." *Journal of Computational physics* 104.1 (1993): 86-92.

Firey, Walter. *Man, Mind, and Land: a Theory of Resource Use*. Glencoe, Ill., Free Press, 1960.

Fried, J.S., Potts, L.D., Loreno, S.M., Christensen, G.A. and Barbour, R.J., 2017a. Inventory-based landscape-scale simulation of management effectiveness and economic feasibility with BioSum. *Journal of Forestry*, *115*(4), pp.249-257.

Fried, J.S., Jain, T.B., Loreno, S., Keefe, R.F., and Bell, C.K., 2017b. A framework for evaluating forest restoration alternatives and their outcomes, over time, to inform monitoring: bioregional inventory originated simulation under management. In Keyser, C.E. and T.L. Keyser, eds. Proceedings of the 2017 Forest Vegetation Simulator (FVS) Virtual e-Conference. E-Gen. Tech. Rep. SRS-224. Asheville, NC: U.S. Department of Agriculture Forest Service, Southern Research Station. 200 p.

Fried, J.S., Loreno, S., Sharma, B., Starrs, C., and Stewart, W., 2016. Inventory based landscape-scale simulation to assess effectiveness and feasibility of reducing fire hazards and improving forest sustainability in California with BioSum. Alternative and Renewable Fuel and Vehicle Technology Program - Final Project Report to the California Energy Commission. 214 p.

Jain, T.B., Battaglia, M.A., Han, H.S., Graham, R.T., Keyes, C.R., Fried, J.S. and Sandquist, J.E., 2012. A comprehensive guide to fuel management practices for dry mixed conifer forests in the northwestern United States.

Jain, T.B., J.S. Fried, R.F. Keefe, S. Loreno, C. Bell. 2017. Evaluating cost-effectiveness of multi-purpose fuel treatments in western mixed-conifer forests considering hazard, risk, longevity and co-benefits: Final Project Report to Joint Fire Sciences Program. [draft in preparation].

North, M., Brough, A., Long, J., Collins, B., Bowden, P., Yasuda, D., Miller, J. and Sugihara, N., 2015. Constraints on mechanized treatment significantly limit mechanical fuels reduction extent in the Sierra Nevada. Journal of Forestry, 113(1), pp.40-48.

Oregon Department of Forestry. 2017. Yearly pond values [online]. Available from http://www.oregon.gov/ODF/Working/pages/TimberSales.aspx [Accessed 27 July, 2017]

Toman, E., Stidham, M., Shindler, B. and McCaffrey, S., 2011. Reducing fuels in the wildland–urban interface: community perceptions of agency fuels treatments. International Journal of Wildland Fire, 20(3), pp.340-349.

Wildland Fire Science, Earth Resources Observation and Science Center. 2016. LANDFIRE Fire Regime Groups, 2014. United States Geological Survey. Accessed 11 Sept. 2017

CONCLUSIONS

The IWFL Biochar project sought to examine the potential for a “win-win-win” scenario, and the initial results presented here indicate that one of those wins is occurring, and the other could occur, but is by no means a sure thing.

The studied tethered cut-to-length system presented a dramatic reduction in the cost of a fuels reduction treatment when compared to the best available estimates for cable logging in a similar scenario. These findings were largely consistent with the few existing comparable studies (McIver et al. 2003, Flint 2013) and, when viewed through the prism of current log prices, suggest that many steep, overstocked areas could break even or generate revenue through treatment with tethered systems.

Despite these findings, and our best efforts, much work remains to be done in describing tethered systems. Conducting the study on a single treatment area allowed for great detail in data collection but significantly limited our ability to assess elements of stand to stand variation, such as soil strength and stability, or treatment to treatment variation, such as retention tree spacing or operator effect. Likewise, the lack of extreme slope on site (where a maximum of 70% slope was operated on) suggests that our description of the harvesting system is fundamentally limited. Within the data used by Neo-Processor, the Upper Klamath contains only approximately 1,500 hectares on slopes greater than seventy percent. This may indicate that potential behavioral changes above the seventy percent threshold are largely unimportant for the region due to the limited number of acres affected, but the area estimate itself may be a significant underestimation, due to how high that slope threshold is and how coarse the underlying FIA data is as a whole. We have observed tethered systems being used on significantly steeper slopes in the Oregon Coast range, but we have no assurances that there are not significant differences in machine or operator behavior, or productivity, beyond a certain slope. It is also entirely possible that if such a “behavior breakpoint” exists, it will vary by operator experience and personal proclivity. Just as some are ill-suited to walking a high girder in construction, some will certainly be ill-suited to driving up and down a 90% slope.

Post-treatment stand conditions also bear further investigation. Damage to residual trees is a perennial concern in any silvicultural treatment that requires tree retention (Hartsough, 2003). Cut-to-length systems have been shown to cause relatively little damage to residual trees in thinning operations (Bettinger and Kellogg, 1993) but it is possible that the reduction in mobility and increase in operator discomfort at higher slopes could make accurate directional felling more difficult and, in doing so, increase the damage done to the residual trees. Soil damage, in terms of both erosion and compaction, could also prove problematic and needs to be conclusively studied. High slopes are well known to create higher erosion rates from overland flow, which tend to increase when the soil is disturbed or the vegetative cover removed (Reynard 1997). The use of brush mats has been shown to significantly decrease the pressure a machine exerts on the soil (Labelle and Jaeger, 2012), but whether they aid, hinder, or have no effect on erosion over the long term is, as yet, unknown.

Of all the tasks that remain after this study, examining the effects of brush mats on fuel loading, fire risk, and fire hazard may be the most pressing. The existence of brush mats is a key delineation between cut-to-length and whole-tree treatments, the former leaving the vast majority of branches out on the stand (Figure 4.1), while the latter generally collects many of the branches at the landing as a result of delimbing and bucking the tree there. Such collection can allow for easier disposal (such as via a pile and burn). Given that we know the tops and branches remain on site, several potential scenarios could exist:

- The increased surface fuel loading coupled with fuel compaction and soil intermixing the likelihood of high residency-time ground fires, potentially increasing tree mortality through root death
- The increased fuel loading of primarily smaller diameter fuels could lead to flashier, more intense ground fires, potentially increasing tree mortality through girdling
- The mastication caused by use in the brush mat could lead to increased surface area and decay rates, with a period of higher risk than normal risk as materials initially dry followed by a period of lower risk as fuel loading drops more rapidly than if the material had been left intact
- The intermixing of branch mat material with soil could effectively act as a fire retardant, reducing fire risk and hazard while aiding in nutrient cycling by retaining

the vast majority of needles on site (albeit in a more linear arrangement than they existed in pre-treatment).

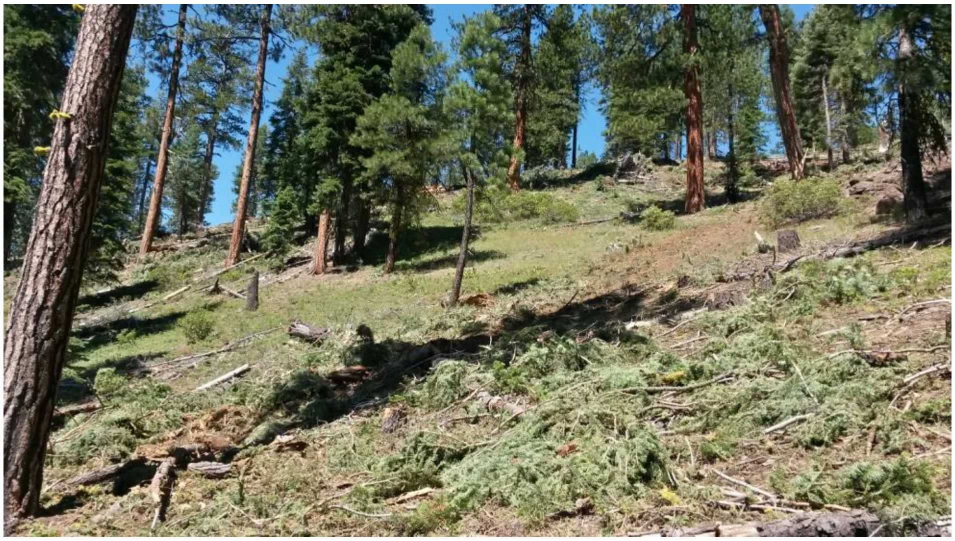


Figure 4.1: Post-treatment surface fuel loading on Pilot Project Unit 10

Any of these scenarios, or some combination of them, could well prove to be true. But no amount of first-principles postulation can prove which is correct. For that, real observations and real data from controlled studies is required.

Despite these unknowns, the tethered cut-to-length technology is already in use, and all evidence indicates it can cost efficiently treat areas that may otherwise have been economically infeasible to treat with other existing systems. From the perspective of the IWFL Biochar Project’s goals, this appears to be a clear win occurring on the landscape right now.

Biochar production has the potential to be a winning proposition as well. We found that biochar payments had a significant, positive effect on the economic feasibility of individual treatments, and that federal fuels treatments were more than capable of supplying the desired 45,360 bone dry tonnes annually, even under the most restrictive scenarios tested. However, those findings do come with caveats. Given the twenty-year planning horizon and single-entry nature of the analysis, we cannot assess or estimate the long-term sustainability of repeated mechanical entry in terms of cumulative impacts or the availability of material large enough to generate saw log or biochar feedstock material on second, third, or further entries. Likewise, we did not assess or estimate the impacts of being restricted to less impactful treatment options (lower volumes removed). As our results show, the more material a treatment removes, the more successful it will be at reducing fire hazard as measured by Composite Resistance Score. But it is often also true that the more material a treatment removes, the harder it is for that treatment to find social acceptance. It is entirely possible that with a more limited set of less impactful treatments the potential feedstock supply would drop below desired levels.

The likelihood of a large-scale market developing remains relatively uncertain. The gains a farmer could expect to see from biochar application are still under investigation, so too is the price a farmer is willing to pay to a biochar processing facility, and (by extension) the price a processing facility is willing to pay for feedstock. The study of biochar as a whole has exploded in recent years, with a cursory Google Scholar search revealing over 1,600 articles with "biochar" in the title from 2017 alone. For contrast, a similar search for "cable logging" yields two articles for the same time period. These findings have been mixed, likely due to the extremely high levels of variation in biochar parent materials, production processes, and application methods leading to wildly different products and practices being contained under the same general banner. A meta-analysis on efforts to study applying biochar to forest soils by Thomas and Gale (2015) found an average biomass increase in biomass of 41%. By contrast, Fornes et al. (2017) found a significant decrease in tomato yield associated with the application of forest-origin biochar. Gaskin et al. (2010) observed yield decreases one year and increases the next, both associated with the same pine-chip biochar and crop. The IWFL Biochar Project's own efforts to characterize biochar made from the kind of conifer residuals expected from fuels reduction treatments is ongoing.

From the perspective of the IWFL Biochar Project's goals this is a win *in potentia*. The development of a biochar market near to the dry, overstocked forests that could benefit from an outlet for small diameter material is by no means certain, but it does have a clear potential to improve the economic outcomes of treatment. Where costs are high, yields are low, and/or tethered systems cannot be adopted due to concerns over erosion or fuel loading, that improvement could prove critical to the economic feasibility of treatment.

Commented [JS4]: Josh, not sure what this means. Was part of the sentence left out?

Commented [JP5R4]: Does this help?

BIBLIOGRAPHY

Agnieszka, M.J., 2016. Biochar as a soil amendment. *Soil Science Annual*, *67*(3), pp.151-157.

Amishev, D., Evanson, T. and Raymond, K., 2009. Felling and Bunching on Steep Terrain–A Review of the Literature. FFR Technical Note, 1(7), pp.1-10.

Barrett, S., Havlina, D., Jones, J., Hann, W.J., Frame, C., Hamilton, D., Schon, K., DeMeo, T., Hutter, L. and Menakis, J., 2010. Interagency Fire Regime Condition Class (FRCC) Guidebook, version 3.0. USDA Forest Service, US Department of the Interior, and The Nature Conservancy. http://www. frcc. gov.

Bell, J.F. and Dilworth, J.R., 1988. Log scaling and timber cruising. OSU Book Stores. Inc., Corvallis, Oregon.

Bettinger, P. and Kellogg, L.D., 1993. Residual stand damage from cut-to-length thinning of second-growth timber in the Cascade Range of western Oregon. *Forest Products Journal*, *43*(11, 12), p.59.

Brassard, P., Godbout, S. and Raghavan, V., 2016. Soil biochar amendment as a climate change mitigation tool: Key parameters and mechanisms involved. *Journal of environmental management*, *181*, pp.484-497.

Brinker, R.W., Kinard, J., Rummer, R., and Lanford, B.L., 2002. Machine rates for selected forest harvesting machines.

Brunson, M.W. and Shindler, B.A., 2004. Geographic variation in social acceptability of wildland fuels management in the western United States. Society and Natural Resources, 17(8), pp.661-678.

Burnham, K.P. and Anderson, D.R., 2002. Model selection and multimodel inference: a practical information-theoretic approach. Springer.

Busenberg, G., 2004. Wildfire management in the United States: the evolution of a policy failure. *Review of policy research*, *21*(2), pp.145-156.

Christian, J.A. 2014. Oregon Department of Forestry logging cost update. Master of Forestry project, Department of Forest Engineering, Resources, and Management, Oregon State University, Corvallis, Oregon.

Denardo, Eric V. *Dynamic programming: models and applications*. Courier Corporation, 2012.

Dueck, Gunter. "New optimization heuristics: The great deluge algorithm and the record-to-record travel." *Journal of Computational physics* 104.1 (1993): 86-92.

Eisenhauer, J.G., 2003. Regression through the origin. Teaching statistics, 25(3), pp.76-80.

Firey, Walter. *Man, Mind, and Land: a Theory of Resource Use*. Glencoe, Ill., Free Press, 1960.

Flint, B.R., 2013. Analysis and operational considerations of biomass extraction on steep terrain in western Oregon. Master of Science, Oregon State University, Corvallis, OR.

Fonseca, M.A., 2005. The measurement of roundwood: methodologies and conversion ratios. CABI.

Fornes, F. and Belda, R.M., 2017. Acidification with nitric acid improves chemical characteristics and reduces phytotoxicity of alkaline chars. *Journal of environmental management*, *191*, pp.237-243.

Fornes, F., Belda, R.M., Fernández de Córdova, P. and Cebolla-Cornejo, J., 2017. Assessment of biochar and hydrochar as minor to major constituents of growing media for containerized tomato production. *Journal of the Science of Food and Agriculture*.

Fried, J.S., Potts, L.D., Loreno, S.M., Christensen, G.A. and Barbour, R.J., 2017a. Inventory-based landscape-scale simulation of management effectiveness and economic feasibility with BioSum. *Journal of Forestry*, *115*(4), pp.249-257.

Fried, J.S., T.B. Jain, S. Loreno, R.F. Keefe, and C.K. Bell. 2017b. A framework for evaluating forest restoration alternatives and their outcomes, over time, to inform monitoring:

bioregional inventory originated simulation under management. In Keyser, C.E. and T.L. Keyser, eds. Proceedings of the 2017 Forest Vegetation Simulator (FVS) Virtual e-Conference. E-Gen. Tech. Rep. SRS-224. Asheville, NC: U.S. Department of Agriculture Forest Service, Southern Research Station. 200 p.

Fried, J.S., Loreno, S., Sharma, B., Starrs, C., and Stewart, W., 2016. Inventory based landscape-scale simulation to assess effectiveness and feasibility of reducing fire hazards and improving forest sustainability in California with BioSum. Alternative and Renewable Fuel and Vehicle Technology Program - Final Project Report to the California Energy Commission. 214 p.

Garber, Sean M., and Douglas A. Maguire. "Modeling stem taper of three central Oregon species using nonlinear mixed effects models and autoregressive error structures." *Forest Ecology and Management* 179.1 (2003): 507-522.

Gaskin, J.W., Speir, R.A., Harris, K., Das, K.C., Lee, R.D., Morris, L.A. and Fisher, D.S., 2010. Effect of peanut hull and pine chip biochar on soil nutrients, corn nutrient status, and yield. *Agronomy Journal*, *102*(2), pp.623-633.

Ghaffariyan, M.R., Acuna, M. and Ackerman, P., 2012. Review of new ground-based logging technologies for steep terrain. CRC for Forestry Bulletin, pp.1-5.

Gullberg, T., 1995. Evaluating operator-machine interactions in comparative time studies. Journal of Forest Engineering, 7(1), pp.51-61.

Hann, David W. "Revised volume and taper equations for six major conifer species in southwest Oregon." *Department of Forest Engineering, Resources, and Management. Oregon State University, Corvallis, Oregon. Forest Biometrics Research Paper 2* (2016).

Hartsough, B., 2003. Economics of harvesting to maintain high structural diversity and resulting damage to residual trees. *Western Journal of Applied Forestry*, *18*(2), pp.133-142.

Jain, T.B., Battaglia, M.A., Han, H.S., Graham, R.T., Keyes, C.R., Fried, J.S. and Sandquist, J.E., 2012. A comprehensive guide to fuel management practices for dry mixed conifer forests in the northwestern United States.

Jain, T.B., J.S. Fried, R.F. Keefe, S. Loreno, C. Bell. 2017. Evaluating cost-effectiveness of multi-purpose fuel treatments in western mixed-conifer forests considering hazard, risk, longevity and co-benefits: Final Project Report to Joint Fire Sciences Program. [draft in preparation].

Keegan III, C.E., Morgan, T.A., Blatner, K.A. and Daniels, J.M., 2010. Trends in lumber processing in the Western United States. Part I: Board foot Scribner volume per cubic foot of timber. Forest products journal, 60(2), pp.133-139.

Kellogg, L.D. and Bettinger, P., 1994. Thinning productivity and cost for a mechanized cut-to-length system in the Northwest Pacific Coast region of the USA. Journal of Forest Engineering, 5(2), pp.43-54.

Labelle, E.R. and Jaeger, D., 2012. Quantifying the use of brush mats in reducing forwarder peak loads and surface contact pressures. *Croatian Journal of Forest Engineering: Journal for Theory and Application of Forestry Engineering*, *33*(2), pp.249-274.

Lindroos, O. and Cavalli, R., 2016. Cable yarding productivity models: a systematic review over the period 2000–2011. International Journal of Forest Engineering, 27(2), pp.79-94.

Long, C.J., Whitlock, C., Bartlein, P.J. and Millspaugh, S.H., 1998. A 9000-year fire history from the Oregon Coast Range, based on a high-resolution charcoal study. *Canadian Journal of Forest Research*, *28*(5), pp.774-787.

Lotan, J. 1976. Cone serotiny - fire relationships in lodgepole pine. *In*: Tall Timbers Fire Ecology Conference Proceedings 14, Tall Timbers Research Center, Tallahassee, FL. pp. 267-278

McIver, J.D., Adams, P.W., Doyal, J.A., Drews, E.S., Hartsough, B.R., Kellogg, L.D., Niwa, C.G., Ottmar, R., Peck, R., Taratoot, M. and Torgersen, T., 2003. Environmental effects and economics of mechanized logging for fuel reduction in northeastern Oregon mixed-conifer stands. Western Journal of Applied Forestry, 18(4), pp.238-249.

Motulsky, H. and Christopoulos, A., 2004. Fitting models to biological data using linear and nonlinear regression: a practical guide to curve fitting. Oxford University Press.

North, M., Brough, A., Long, J., Collins, B., Bowden, P., Yasuda, D., Miller, J. and Sugihara, N., 2015. Constraints on mechanized treatment significantly limit mechanical fuels reduction extent in the Sierra Nevada. Journal of Forestry, 113(1), pp.40-48.

Olmstead, S., Kousky, C. and Sedjo, R., 2012. Wildland Fire Suppression and Land Development in the Wildland/Urban Interface.

Oregon Department of Forestry. 2017. Yearly pond values [online]. Available from http://www.oregon.gov/ODF/Working/pages/TimberSales.aspx [accessed 27 July, 2017]

Pyne, S.J., 2011. *Fire: a brief history*. University of Washington Press.

Renard, K.G., 1997. Predicting soil erosion by water: a guide to conservation planning with the revised universal soil loss equation (RUSLE).

Ryan, K.C., Knapp, E.E. and Varner, J.M., 2013. Prescribed fire in North American forests and woodlands: history, current practice, and challenges. Frontiers in Ecology and the Environment, 11(s1).

Sessions, J., Leshchinsky, B., Chung, W., Boston, K., and Wimer, J. 2017. Theoretical Stability and Traction of Steep Slope Tethered Feller-Bunchers. Forest Science 63(2):192-200.

Shindler, B. and Toman, E., 2003. Fuel reduction strategies in forest communities: A longitudinal analysis of public support. *Journal of Forestry*, *101*(6), pp.8-15.

Stephens, S.L. and Ruth, L.W., 2005. Federal Forest-Fire Policy in the United States. Ecological applications, 15(2), pp.532-542.

Thomas, S.C. and Gale, N., 2015. Biochar and forest restoration: a review and meta-analysis of tree growth responses. *New Forests*, *46*(5-6), pp.931-946.

Todd, S.K. and Jewkes, H.A., 2006. Wildland fire in Alaska: a history of organized fire suppression and management in the last frontier.

Toman, E., Stidham, M., Shindler, B. and McCaffrey, S., 2011. Reducing fuels in the wildland–urban interface: community perceptions of agency fuels treatments. International Journal of Wildland Fire, 20(3), pp.340-349.

Toman, E., Shindler, B., McCaffrey, S. and Bennett, J., 2014. Public acceptance of wildland fire and fuel management: Panel responses in seven locations. *Environmental management*, *54*(3), pp.557-570.

United States Department of Agriculture. 2017. OR.accdb. Available from https://apps.fs.usda.gov/fia/datamart/datamart_access.html [accessed 10 May, 2017]

Visser, R. and Stampfer, K., 2015. Expanding ground-based harvesting onto steep terrain: a review. Croatian journal of forest engineering, 36(2), pp.321-331.

Wildland Fire Science, Earth Resources Observation and Science Center. 2016. LANDFIRE Fire Regime Groups, 2014. United States Geological Survey. Accessed 11 Sept. 2017

Wright, C.S. and Agee, J.K., 2004. Fire and vegetation history in the eastern Cascade Mountains, Washington. *Ecological Applications*, *14*(2), pp.443-459.

APPENDIX - NEO-PROCESSOR DOCUMENTATION

A Word on Documentation Structure

The documentation for each program is separated into four sections:

- The Model Flow section describes inputs, outputs, assumptions, and the rules the model uses to generate moves and find solutions.
- The User Control section describes variables and settings used to quickly and easily modify individual runs.
- The Object List section describes how data is stored in the model, including their units and formatting. Lastly,
- The Function List describes how objects and variables are used by the individual functions that, taken together, comprise each program.

The Model Flow section is intended to explain *what* each program does to someone who is familiar with heuristics and other optimization techniques but might be unfamiliar with Python or other similar programming languages. By contrast, the User Control, Object List and Function List describe *how* each program does what it does and is meant to guide those who might want to examine, use, or modify the code on their own.

Throughout the User Control, Object List, and Function List, important names are color coded:

- Green names are used when referencing objects such as fragments or cut trees.
- Blue names are used when referencing functions such as gendib or gensol.
- Orange names are used when referencing variables. This includes singular values such as flood, list names like frl, or dictionary names like rxf.

All objects and functions are defined in their respective sections in order of use, with the first objects/functions called within the program being the first defined.

A Word on Lists and Dictionaries

While a working knowledge of the Python programming language is not necessary to understand *what* each program does, two things are critical to understanding *how* they do what they do:

1. Lists are ordered sets of information with data stored at indexes. Individual values in a list are referenced by their index, with the first value in the list at [0] (index zero), the second at [1] (index one) and so on. Most objects in RxEvaluator and Neo-Processor are lists with information stored at specific indexes.

2. Dictionaries are sets of information that work on key/value pairs. When a given key is called within a dictionary it returns the value associated with that key. Most objects in RxEvaluator and Neo-Processor are stored in *nested* dictionaries. For example, fragment outputs are stored in rxf [standid] [rx] [ac], where rxf is a dictionary keyed by standid (the unique identifier for the stand the fragment is a part of), standid is also a dictionary keyed by rx (the prescription code), and rx is also a dictionary keyed by ac (the number of acres in the fragment).

A Word on Species Codes and Material Definitions

Throughout Neo-Processor two distinct systems of species identification are used:

- FIA/FVS species codes, and
- Tree Eater species codes.

FVS species codes will generally correspond to FIA species codes, save for when species occur in a stand without being present in the variant assigned to that stand. When that happens FVS assigns the tree in question to the other softwood code (298) or other hardwood code (998) as appropriate. Tree Eater species codes are used to identify and (in some cases) group commercial species for analysis, using the same prices and taper equations for each species in the group.

Likewise, throughout Neo-Processor two different types of material are produced and tracked:

- Saw-quality material is made up of the bole wood of commercial species that can be cut into logs between eight and twenty eight feet in length and between six and twenty four inches in top-end diameter inside bark.

- Biochar feedstock is made up of the bole wood of all species that can be cut into logs of eight feet or more in length with a top-end diameter inside bark of four inches or greater.

Neo-Processor does not track the material from tops of less than four inches in diameter or branches. Neo-Processor was designed explicitly for use with cut-to-length systems where such material is generally used to make brush mats and masticated, compacted, and/or mixed with topsoil as a result.

RX Evaluator

Model Flow

Rx Evaluator is the first of two programs designed to substitute for the Processor module in BioSum, with the other being Neo-Processor. The decision was made to split Rx Evaluator and Neo-Processor to save on overall processing time. Rx Evaluator is a custom script written in Python 2.7 and executed in the Spyder integrated development environment. Rx Evaluator requires the user to provide values for five different user controls (as described in the Rx Evaluator User Controls) and a number of supplemental inputs as tab-delimited text files.

Rx Evaluator requires a host of inputs, some are provided along with the script, and others must be supplied by the user. Both types of inputs are stored externally as tab-delimited text files. The provided inputs are:

- The parameters for calculating diameter inside bark, stored as dibparms.txt in the rxe_parameters subfolder
- The parameters for calculating green weights, stored as gwtparms.txt in the rxe_parameters subfolder
- The parameters for calculating predicted volume mortality, stored as mrtparms.txt in the rxe_parameters subfolder, and
- The Westside Scribner board foot volumes for logs between eight and twenty eight feet in length and between zero and twenty four inches in top-end diameter inside bark, stored as scribtable.txt in the rxe_parameters subfolder.

The user supplied inputs are:

- The facility capability data, stored as fcap.txt in the rxe_parameters subfolder
- The transport costs from each stand to each facility, stored in tcost.txt in the rxe_parameters subfolder
- The average yarding distance for each stand, stored as yard.txt in the rxe_parameters subfolder, and
- Four files for each prescription to be analyzed, saved as XXXc.txt, XXXl.txt, XXXp.txt, and XXXs.txt in the rxe_FVSinputs subfolder, where XXX is the prescription code. These files are generated from FVS output databases using SQL queries (provided in the Rx Evaluator User Control section).

All treatments use a three digit naming convention, where the first two digits are the treatment number and the last digit is the treatment period. Treatment code 999 is reserved and should always be used for the no-action alternative.

Once all inputs are provided and the script is run, Rx Evaluator performs a transportation analysis, identifying and storing the lowest cost route from each stand to each facility for each material type and (for saw-quality material) Tree Eater species code. After the lowest cost routes are identified, each prescription is evaluated in turn by passing through three different modules, Tree Eater, Accountant, and Fire Eater.

The Tree Eater module uses the cut list for a given prescription to perform three key tasks:

- Optimally bucking each tree in the cut list capable of producing saw-quality material
- Calculating the volume of biochar feedstock produced by each tree in the cut list, and
- Using the outputs from that bucking and calculation to determine the expected yields per acre for each stand treated by that prescription.

By default, Tree Eater has six species codes (Table A.1) comprised of twelve individual species, selected from the species present in the Upper Klamath basin. Species were selected for inclusion in Tree Eater by only if they met each of three criteria:

- Being a species acceptable to local mills for producing traditional, high value wood products (not pulp or biochar)
- Being a species likely to be removed during a treatment aimed at fuels reduction, and
- Being a species with available price and sort data.

Table A.1 – Default Tree Eater Species Codes

In FVS / FIA	
FIA Code	Common Name
🟩 11	Pacific silver fir
🟩 15	White fir
🟩 17	Grand fir
🟩 19	Subalpine fir
🟩 20	Red fir
🟩 21	Shasta fir
🟧 64	Western juniper
🟩 81	Incense cedar
🟧 93	Engelmann spruce
🟧 101	Whitebark pine
🟩 108	Lodgepole pine
🟩 116	Jeffrey pine
🟩 117	Sugar pine
🟧 119	Western white pine
🟩 122	Ponderosa pine
🟩 202	Douglas-fir
🟧 264	Mountain hemlock
🟧 321 / 998	Rocky mountain maple
🟧 431	Chinkapin
🟧 475	Mountain mahogany
🟧 746	Quaking aspen
🟧 763	Chokecherry
🟧 815	Oregon white oak
🟧 818	California black oak

Species Status for Bucking	
🟩	Commercial Species
🟧	Non-commercial Species

In Tree Eater	
TE Code	Common Name
1	Douglas-fir
2	Incense cedar
3	Jeffrey pine
3	Ponderosa pine
4	Sugar pine
5	Lodgepole pine
6	Pacific silver fir
6	White fir
6	Grand fir
6	Subalpine fir
6	Red fir
6	Shasta fir

Table A.2: Default Log Prices ($/MBF) by Sort and Species Aggregate

TE Code	Aggregate	Sort	2015.3	2015.4	2016.1	2016.2	Ave.
1	Douglas-fir	6"-8"	$465.00	$475.00	$490.00	$495.00	$481.25
		8"-14"	$510.00	$520.00	$525.00	$500.00	$513.75
		14"-22"	$520.00	$535.00	$530.00	$520.00	$526.25
		22"+	$525.00	$545.00	$535.00	$525.00	$532.50
2	Incense cedar	6"-8"	$650.00	$600.00	$650.00	$640.00	$635.00
		8"-14"	$650.00	$600.00	$650.00	$640.00	$635.00
		14"-22"	$650.00	$600.00	$650.00	$640.00	$635.00
		22"+	$650.00	$600.00	$650.00	$640.00	$635.00
3	Ponderosa pine	6"-8"	$285.00	$270.00	$265.00	$310.00	$282.50
		8"-14"	$320.00	$315.00	$305.00	$325.00	$316.25
		14"-22"	$355.00	$350.00	$335.00	$340.00	$345.00
		22"+	$390.00	$385.00	$365.00	$370.00	$377.50
4	Sugar pine	6"-8"	$285.00	$280.00	$275.00	$260.00	$275.00
		8"-14"	$305.00	$295.00	$300.00	$280.00	$295.00
		14"-22"	$335.00	$360.00	$320.00	$305.00	$330.00
		22"+	$365.00	$335.00	$345.00	$335.00	$345.00
5	Lodgepole pine	CR	$325.00	$320.00	$360.00	$365.00	$342.50
6	True fir	6"-8"	$380.00	$385.00	$370.00	$375.00	$377.50
		8"-14"	$410.00	$390.00	$380.00	$385.00	$391.25
		14"-22"	$415.00	$405.00	$410.00	$410.00	$410.00
		22" +	$420.00	$415.00	$415.00	$415.00	$416.25

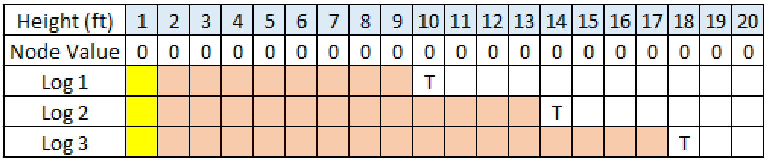
By default, price and sort information (Table A.2) is built to reflect the Klamath Unit average of the four most recent quarters available on the State of Oregon’s Open Data website at the time of Neo-Processor’s design, specifically quarters three and four of 2015 and quarters one and two of 2016. Five of the six Tree Eater species are priced with the same diameter class breakdown, but lodgepole pine has only a single “camp run” value. A camp run log is any log of greater than cull quality (Bell and Dillworth, 1988).

Tree Eater’s optimal bucking process uses a forward reaching algorithm (Denardo 2012) to find the best log or combination of logs that could come from any given tree. It does this by breaking each tree down into one foot increments called “nodes.” Starting from an assumed one foot stump, Tree Eater moves up the tree node by node, identifying and evaluating all logs that could originate from each node and saving the value those logs to the last node of that log (the trim). As Tree Eater moves up the tree, it begins comparing different paths to the same node, saving the best it finds each time. After each node capable of creating a log has been evaluated, the highest value node will contain the optimal value for the tree.

Tree Eater only allows logs of eight, twelve, sixteen, twenty, twenty four, or twenty eight feet in length and assumes one additional foot per log of trim. By default, Tree Eater only generates nodes until the diameter inside bark for a node drops below four inches and will only attempt to create saw-quality logs while the top-end diameter inside bark is six inches or greater (this does not include trim).

To illustrate this process, assume Tree Eater finds a tree in the cut list with a valid Tree Eater species code, a diameter inside bark that falls below six inches twenty feet up the bole, and falls below four inches twenty nine feet up the bole. It would generate twenty eight nodes (Figure A.1), and evaluate them from the bottom up. From the first node (in yellow) three logs (in brown) with one foot of trim (t) would be possible, of eight feet, twelve feet, and sixteen feet in length.

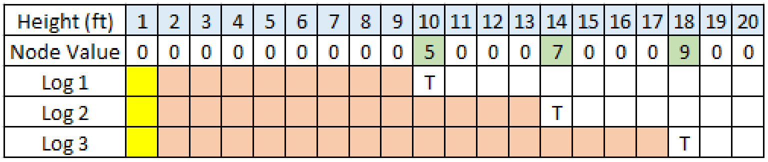
Figure A.1: Forward Reaching Example Part 1



Height (ft)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Node Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Log 1	🟨	🟫	🟫	🟫	🟫	🟫	🟫	🟫	🟫	T										
Log 2	🟨	🟫	🟫	🟫	🟫	🟫	🟫	🟫	🟫	🟫	🟫	🟫	🟫	T						
Log 3	🟨	🟫	🟫	🟫	🟫	🟫	🟫	🟫	🟫	🟫	🟫	🟫	🟫	🟫	🟫	🟫	🟫	T		

Let's further assume that an eight foot log is worth five dollars, a twelve foot log is worth seven dollars, and a sixteen foot log is worth nine dollars, and save the value of those logs to their last node (the trim, Figure A.2).

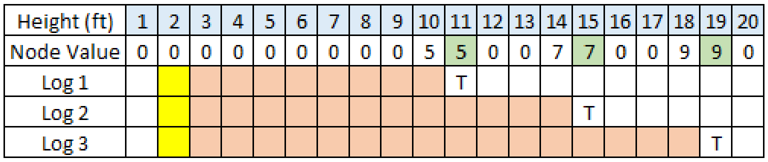
Figure A.2: Forward Reaching Example Part 2



Height (ft)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Node Value	0	0	0	0	0	0	0	0	0	5	0	0	0	7	0	0	0	9	0	0
Log 1	🟨	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	T										
Log 2	🟨	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	T						
Log 3	🟨	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	T		

Once those values are saved Tree Eater moves onto the next node and the process repeats (Figure A.3).

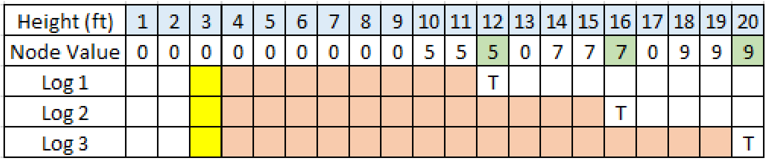
Figure A.3: Forward Reaching Example Part 3



Height (ft)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Node Value	0	0	0	0	0	0	0	0	0	5	5	0	0	7	7	0	0	9	9	0
Log 1		🟨	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	T									
Log 2		🟨	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	T					
Log 3		🟨	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	T	

Since the diameter inside bark drops below six inches at node twenty, node three is the last node with three possible logs (Figure A.4).

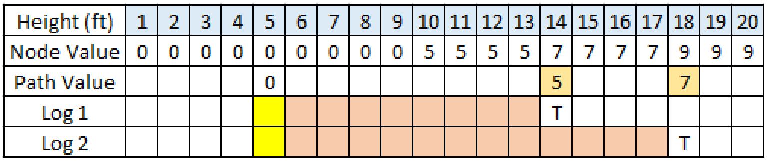
Figure A.4: Forward Reaching Example Part 4



Height (ft)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Node Value	0	0	0	0	0	0	0	0	0	5	5	5	0	7	7	7	0	9	9	9
Log 1			🟨	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	T								
Log 2			🟨	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	T				
Log 3			🟨	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	T

At node five, cells begin to overlap, testing different paths to the same node against each other (Figure A.5).

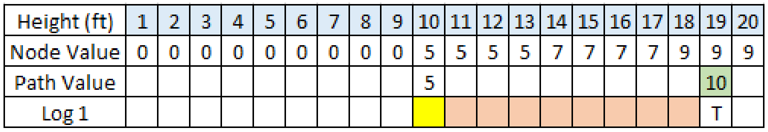
Figure A.5: Forward Reaching Example Part 5



Height (ft)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Node Value	0	0	0	0	0	0	0	0	0	5	5	5	5	7	7	7	7	9	9	9
Path Value					0									5				7		
Log 1					🟨	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	T						
Log 2					🟨	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	T		

In this case, the potential logs ending at node fourteen and eighteen are rejected because the model has already found a higher value path to those nodes. This changes at node ten (Figure A.6).

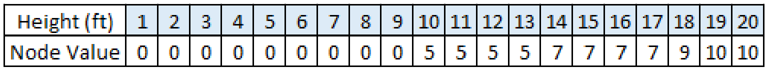
Figure A.6: Forward Reaching Example Part 6



Height (ft)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Node Value	0	0	0	0	0	0	0	0	0	5	5	5	5	7	7	7	7	9	9	9
Path Value										5									10	
Log 1										🟨	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	T	

At node ten the path value is no longer just the value of the current log, it’s the value of the current log and the previous log. The two eight foot logs earn more than the one sixteen foot log, so this new value is saved. The same thing happens at node eleven, and since node eleven is the last node capable of making a saw-quality log, bucking stops and the best value for the tree ($10) is saved (Figure A.7).

Figure A.7: Forward Reaching Example Part 7

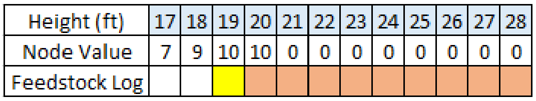


Height (ft)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Node Value	0	0	0	0	0	0	0	0	0	5	5	5	5	7	7	7	7	9	10	10

Once the optimal value for the tree is found, the model determines how much (if any) biochar feedstock can be produced from the remainder. It does this by identifying the lowest node containing the optimal value and measuring the distance from that point to the last node. If that distance is eight feet or more, a feedstock log of that length is produced.

To go back to the previous example, there are two nodes after bucking (nineteen and twenty) that contain the optimal value. Node nineteen is the lowest. The diameter inside bark falls below four inches at twenty nine feet up the bole, so node twenty eight is the last node in the tree (Figure A.8).

Figure A.8: Forward Reaching Example Part 8



Height (ft)	17	18	19	20	21	22	23	24	25	26	27	28
Node Value	7	9	10	10	0	0	0	0	0	0	0	0
Feedstock Log			🟨	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧

The distance between the lowest optimal node and the last node is nine feet, so a nine foot feedstock log is created. For trees of non-commercial species, the entire volume of the tree to a four inch top is used as biochar feedstock. For trees of commercial species that are incapable of making a saw-quality log but capable of making a feedstock log, the entire volume of the tree to a four inch top is used as biochar feedstock.

As each cut tree is processed, Tree Eater calculates the per acre revenue, saw-quality volume and green weight, and biochar feedstock volume and weight associated with that cut tree and adds those yields to running totals for that tree's stand. After the last cut tree is processed, those per acre yields are passed on to the Accountant module.

The Accountant module takes in the per acre yields from Tree Eater and uses them to calculate per acre costs as a function of the machine time required to handle the volume removed. It does this by calculating the cost of the harvest system and the cost to load and transport all saw-quality material and biochar feedstock produced by the treatment. Accountant assumes a tethered cut-to-length harvest system will be used for all treatments, with a harvester felling and bucking and a forwarder yarding and loading the material onto trucks for transport.

Harvester time is calculated as:

$$ht = 1.43tw + 0.25hdi$$

Where ht is the harvester time required in productive machine minutes, tw is the total weight of material handled in green tonnes, and hdi is the distance traveled in meters. hdi is held constant at 531 meters per acre and was calculated as two times the linear distance required to cover one acre with a corridor fifty feet wide. This may overestimate the actual distance required on flat ground where opportunistic wandering may allow the harvester to move from section to section without backtracking. It may also underestimate the actual distance on short, steep corridors where it fails to account for travel time between corridors. The harvester time in minutes is then converted to hours and multiplied by either $278.95 per productive machine hour for tethered operations or $212.47 per productive machine hour for untethered operations, producing a harvester cost per acre.

Forwarding time is calculated as:

ft = 0.05fdi + 2.19sac + 4.06pac on tethered ground, and

ft = 0.01fdi + 2.19sac + 4.06pac on untethered ground

Where ft is the forwarding time required in productive machine minutes, sac is the weight of sawlog material in green tonnes, pac is the weight of feedstock material in green tonnes, and fdi is the distance traveled by the forwarder in meters. fdi is calculated by multiplying the number of bunks per acre by the two times the average yarding distance. The number of bunks per acre is calculated by dividing the total green tonnes per acre by twelve and rounding to the nearest whole number. fdi is also subject to a minimum distance of 473 meters per acre, calculated as two times the linear distance required to cover one acre with a corridor sixty feet wide, with an additional thirty meters added per bunk to account for out of corridor travel to a pile. The forwarding time in minutes is then converted to hours and multiplied by either $198.11 per productive machine hour for tethered operations or $172.22 per productive machine hour for untethered operations, producing a forwarding cost per acre.

Truck loading time is calculated as:

lt = (sac / lsp) + (pac / lpp)

Where lt is the time spent loading in productive machine hours, lsp is the loading rate for saw material in green tonnes per hour and lpp is the loading rate for feedstock material in green tonnes per hour. Loading cost is then calculated as the loading time multiplied by $213.22 ($172.22 per productive machine hour for the forwarder plus $46 per productive hour for the idle truck).

The cost to transport material of a given type and species from each stand to each appropriate facility is calculated externally using the ArcGIS Network Analyst tool. Accountant takes those costs (in round trip dollars per green ton) and uses them to calculate the transport cost for each stand on a piece by piece basis, identifying the minimum cost route for each piece of saw-quality material and/or biochar feedstock and summing them to build the transportation costs for each stand.

Lastly, Accountant sums the per acre harvester, forwarding, loading, and transportation costs into a single per acre treatment cost. It is important to note that fixed costs per entry (such as those associated with mobilization) and any additional per acre costs (such as those associated with post thinning prescribed burns) are not included in Rx Evaluator, but can be added and managed in Neo-Processor.

After all costs, revenues, and yields have been calculated for a prescription, it passes through one final module before the final results are exported. Fire Eater calculates the combined resistance score (Jain *et al*. 2017) for each stand under each prescription. The composite resistance score is comprised of four components:

- Canopy base height / crown connectivity
- Canopy bulk density
- Basal area of fire resistant species, and
- Predicted volume mortality.

Each component is scored on a scale of zero to three points and those scores are summed to provide a composite resistance score between zero and twelve.

The canopy base height / crown connectivity is calculated in one of two ways. For stands with a single stratum (as defined by FVS defaults) Fire Eater uses the crown base height output from the FVS structure class table. For multi-strata stands, Fire Eater uses canopy connectivity, calculated as the distance between tallest tree of the lowest stratum and the lowest tree of the next-to-lowest stratum. Canopy base height and crown connectivity are both scored in the same fashion:

- Zero points for heights / distances of less than seven feet
- One point for heights / distances between seven and twenty feet
- Two points for heights / distances between twenty and thirty feet, and
- Three points for heights / distances of thirty feet or greater.

Canopy bulk density is taken directly from the FVS Potfire tables and is scored as follows:

- Zero points for densities of 0.15 kilograms per cubic meter or greater
- One point for densities between 0.1 and 0.15 kilograms per cubic meter
- Two points for densities between 0.05 and 0.1 kilograms per cubic meter, and

- Three points for densities of 0.05 kilograms per cubic meter or less

The basal area of resistant species is calculated as percent of total (expressed as a fractional value between zero and one). In Rx Evaluator six species are considered to be fire resistant:

- Red fir
- Shasta fir
- Jeffrey pine
- Ponderosa pine
- Sugar pine, and
- Douglas-fir

Total basal area and resistant basal area are calculated using FVS tree list records, with the tree list TPA adjusted to reflect TPA removed during treatment. The DBH of each tree is multiplied by the Forester's Constant to get the basal area for that tree and then added to the running total for the stand and (depending on species) the running total for resistant species in the stand. Resistant basal area fraction (RBA) is scored by assigning:

- Zero points for an RBA of less than 0.25
- One point for an RBA between 0.25 and 0.5
- Two points for an RBA between 0.5 and 0.75, and
- Three points for an RBA of 0.75 or greater.

Predicted volume mortality is the estimated mortality in the event of a fire with six to eight foot flame lengths, as a percent of the total volume. Predicted volume mortality is calculated using parameters provided by the authors of Jain *et al* (2017) by multiplying the volume for each tree by the appropriate predicted mortality parameter, summing the predicted volume mortality, and dividing it by the total volume.

Table A.3 – Predicted Mortality Parameters by Species and DBH Class

	1-4.9"	5-9.9"	10-14.9"	15-20.9"	21-29.9"	30-39.9"	40-999"
White fir	0.99	0.99	0.65	0.46	0.14	0.09	0.065
Grand fir	1	0.9675	0.94	0.7325	0.47	0.47	0.47
Subalpine fir	1	0.9675	0.94	0.7325	0.47	0.47	0.47
Incense cedar	0.95	0.95	0.95	0.5	0.11	0.025	0.03
Engelmann spruce	0.99	0.99	0.955	0.73	0.72	0.655	0.655
Red fir	0.85	0.535	0.295	0.1425	0.09	0.09	0.09
Shasta fir	0.85	0.535	0.295	0.1425	0.09	0.09	0.09
Whitebark pine	1	0.99	0.9075	0.775	0.37	0.18	0.18
Lodgepole pine	1	0.99	0.9075	0.775	0.37	0.18	0.18
Sugar pine	1	1	0.905	0.56	0.2	0.16	0.125
Jeffrey pine	0.8	0.8	0.66	0.4475	0.2025	0.0725	0.06
Ponderosa pine	0.8	0.8	0.66	0.4475	0.2025	0.0725	0.06
Douglas-fir	0.98	0.98	0.705	0.54	0.275	0.305	0.215
Quaking aspen	1	1	1	1	1	1	1
Western juniper	1	0.991	0.951	0.839	0.708	0.596	0.47
Rocky mountain maple	1	0.991	0.951	0.839	0.708	0.596	0.47
Chinkapin	1	0.991	0.951	0.839	0.708	0.596	0.47
Mountain mahogany	1	0.991	0.951	0.839	0.708	0.596	0.47
Chokecherry	1	0.991	0.951	0.839	0.708	0.596	0.47
Oregon white oak	1	0.991	0.951	0.839	0.708	0.596	0.47
California black oak	1	0.991	0.951	0.839	0.708	0.596	0.47
Pacific silver fir	0.997	0.991	0.868	0.607	0.332	0.157	0.093
Western white pine	0.997	0.991	0.868	0.607	0.332	0.157	0.093
Mountain hemlock	0.997	0.991	0.868	0.607	0.332	0.157	0.093

After calculation, the predicted volume mortality is scored with:

- Zero points for predicted volume mortality of greater than 0.98
- One point for predicted volume mortality between 0.98 and 0.7
- Two points for predicted volume mortality between 0.7 and 0.4, and
- Three points for predicted volume mortality of 0.4 or less.

Composite resistance scores are calculated for each of five years of interest, the four treatment years (one, six, eleven, and sixteen) and the last year in the planning horizon (year twenty). In years where treatment occurs the composite resistance score is always calculated for post-treatment conditions.

Once the yields, revenues, costs, and scores are all calculated, Rx Evaluator packages them into an ordered list, with one such list for every valid stand/prescription combination. Rx Evaluator then exports those lists to a tab delimited text file (rxoutputs.txt) to be imported to and analyzed by Neo-Processor.

User Controls

Rx Evaluator has five built in user control variables for quick iteration and fine tuning of output generation (with Rx Evaluator). These variables are found on lines 865 to 874 of the Rx Evaluator script.

Assumed Bark Moisture Content and Assumed Wood Moisture Content
Variable name: mcbark and mcwood
Default value: 0
Appropriate inputs: See below

The assumed moisture content controls are used for calculating the green weight in tons of harvested material and, by extension, the transportation costs for that material. All moisture contents are calculated on a dry weight basis. These controls have three modes of operation:

- When set to zero, all green weights are calculated using the default values included in the FIA species reference tables.
- When set to a negative number, all green weights are calculated as the FIA defaults minus the number specified. The model defaults to a moisture content of 10% if this control would reduce it to less than 10%.
- When set to a positive number, the moisture content for all species is set to that value.

See the calcbarkwt and calcwoodwt functions for a complete description of the dry volume to wet weight equations used by Rx Evaluator.

Slope Breakpoint Control
Variable name: slopebp
Default value: 35
Appropriate inputs: any positive integer

The slope breakpoint control is used to determine whether the tethered or untethered harvest model is invoked for a given stand. When the slope for the stand is equal to or greater than the slope breakpoint, tethered calculations are used. See the costst function for a complete description of the harvest time and cost equations used by Rx Evaluator.

Prescription Control List
Variable name: rxlist
Default value: NA
Appropriate inputs: any number of valid prescription codes as a comma separated list

The prescription control list holds the codes for each prescription being analyzed in the current run. It is highly recommended that the user break the list into separate lines (using a backslash as a line break) for easy inspection.

Facility Disable List
Variable name: fdisl

Default value: []
Appropriate inputs: any number of facility codes as a comma separated list

The facility disable list is a control designed to let the user remove facilities from consideration without removing those facilities from existing input files. It is important to note that this does not actually prevent facilities from being selected, instead setting their transport costs to $999.90 per green ton. This should prevent facility selection in the vast majority of circumstances, but could create anomalous results if most or all of the facilities that handle a given species are disabled in this way.

SQL Queries

Rx Evaluator relies on queried FVS tables to generate key inputs, those queries are listed here for convenience. The results of each query must be saved to an appropriately named text file and stored in the rxe_FVSinputs subfolder.

Cut List Query

Text file: XXXc.txt
Query: SELECT FVS_CutList.Species, FVS_CutList.DBH, FVS_CutList.Ht, FVS_CutList.TruncHt, FVS_CutList.PctCr, FVS_CutList.MDefect, FVS_CutList.TPA, FVS_CutList.MCuFt, FVS_CutList.StandID, FVS_CutList.TreeId, FVS_CutList.Year
FROM FVS_CutList;

Tree List Query

Text file: XXXl.txt
Query: SELECT FVS_TreeList.Species, FVS_TreeList.DBH, FVS_TreeList.Ht, FVS_TreeList.TruncHt, FVS_TreeList.PctCr, FVS_TreeList.MDefect, FVS_TreeList.TPA, FVS_TreeList.MCuFt, FVS_TreeList.StandID, FVS_TreeList.TreeId, FVS_TreeList.Year
FROM FVS_TreeList;

Potfire Query

Text file: XXXp.txt

```
Query: SELECT FVS_PotFire.StandID, FVS_PotFire.Year, FVS_PotFire.Canopy_Ht,
FVS_PotFire.Canopy_Density
FROM FVS_PotFire;
```

Structure Class Query (Fuels Treatments Only)

Text file: XXXs.txt

```
Query: SELECT DISTINCT (FVS_StrClass.StandID) AS Expr1, FVS_StrClass.Year,
FVS_StrClass.Number_of_Strata, FVS_StrClass.Stratum_1_Crown_Base,
FVS_StrClass.Stratum_1_Sm_Ht, FVS_StrClass.Stratum_2_Lg_Ht,
FVS_StrClass.Stratum_2_Sm_Ht, FVS_StrClass.Stratum_3_Lg_Ht
FROM FVS_StrClass
WHERE (((FVS_StrClass.Removal_Code)=1) AND ((FVS_StrClass.Year)<21));
```

Structure Class Query (No-Action Alternative)

Text file: 999s.txt

```
Query: SELECT DISTINCT (FVS_StrClass.StandID) AS Expr1, FVS_StrClass.Year,
FVS_StrClass.Number_of_Strata, FVS_StrClass.Stratum_1_Crown_Base,
FVS_StrClass.Stratum_1_Sm_Ht, FVS_StrClass.Stratum_2_Lg_Ht,
FVS_StrClass.Stratum_2_Sm_Ht, FVS_StrClass.Stratum_3_Lg_Ht
FROM FVS_StrClass
WHERE (((FVS_StrClass.Year)<21));
```

Object List

The objects used by Rx Evaluator can all be classified as external objects, or internal objects. External objects are stored in tab-delimited text files and read into the model during the “Dictionary Loading” phase of operation. Internal objects are created by external objects interacting with functions and user inputs.

External Trees (Table A.4)

Source: Queried FVS tree list and cut list files

Stored externally in: XXXc.txt and XXXl.txt, where XXX is a valid prescription code

Stored internally in: NA

Table A.4: External Tree Fields

Field	Units	Index	Format	Description
Species	NA	0	Int	FVS tree species code
DBH	in	1	Float	Diameter at breast height in inches
Height	ft	2	Float	Height of the tree in feet (estimated for broken tops)
TrunHt	ft	3	Float	Truncated height of a tree with a broken top in feet
PctCr	%	4	Float	Percent crown ration
Mdefect	%	5	Float	Percent defect in merchantable volume
TPA	NA	6	Float	Trees per acre associated with the tree record
MCuFt	ft3	7	Float	Merchantable volume in cubic feet
StandID	NA	8	Str	Stand identifier associated with the tree
TreeID	NA	9	Int	Tree identifier
Year	NA	10	Int	FVS model year

Before they can become cut trees or leave trees, all trees in Rx Evaluator start in the cut list or tree list of FVS outputs. Those lists are queried to generate these external tree records and stored in the XXXc.txt and XXXl.txt respectively, where XXX is a valid prescription code. It is important to note that the merchantable cubic foot volume is measured to a four inch top and not the FVS default six inch top.

Potfire Values (Table A.5)

Source: Queried FVS Potfire tables

Stored externally in: XXXp.txt, where XXX is a valid prescription code

Stored internally in: prx [rx] [standid] [year]

Table A.5: Potfire Value Fields

Field	Units	Index	Format	Description
Rx	NA	0	Int	Prescription code
StandID	NA	1	Str	Stand identifier associated with the record
Year	NA	2	Int	FVS model year
CBH	ft	3	Float	Height to crown base in feet (defunct)
CBD	kg/m^3	4	Float	Crown bulk density in kilograms per cubic meter

Potfire values contain the crown base height and crown bulk density as calculated by the FVS Potfire extension. The height to crown base is vestigial from earlier model versions and no longer used in Rx Evaluator.

Structure Class Inputs (Table A.6)

Source: Queried FVS structure class tables

Stored externally in: XXXs.txt, where XXX is a valid prescription code

Stored internally in: hrx [rx] [standid] [year]

Table A.6 Structure Class Input Fields

Field	Units	Index	Format	Description
Stand	NA	0	Str	Stand identifier associated with the record
Year	NA	1	Int	FVS model year
#Strata	NA	2	Int	The current number of strata in the stand
HCB	ft	3	Int	The height to crown base in feet
SmHt1	ft	4	Int	The height of the shortest tree in stratum one in feet
LgHt2	ft	5	Int	The height of the tallest tree in stratum two in feet
SmHt2	ft	6	Int	The height of the shortest tree in stratum two in feet
LgHt3	ft	7	Int	The height of the largest tree in stratum three in feet

The structure class inputs are used to calculate the structure class values and, by extension, the composite resistance scores for each stand under each prescription. See structure class values and dictionary loading for a full description of how these values are used.

Scribner Values

Source: Bell and Dillworth (1988)

Stored externally in: scribtable.txt

Stored internally in: slist [length] [topdib]

Scriber values hold the Revised Scribner board foot volumes for logs between eight and twenty eight feet in length (in four foot increments) and one and twenty four inches in top diameter inside bark (in one inch increments). These values are used during bucking to determine the volume of individual logs in each cut tree. The length index used to store scribner values is in four foot increments, such that an eight foot log is at [2], a twelve foot log is at [3], and so on. By default, twenty four foot and twenty eight foot logs are not considered during bucking.

DIB Parameters

Source: Garber and Maguire (2003), Hann (2016)

Stored externally in: dibparms.txt

Stored internally in: mdib [tesp]

DIB parameters are an ordered list of parameters used for calculating the diameter inside bark during bucking. The parameters for lodgepole pine are taken from the corrected Garber and Maguire (2003) and the parameters for all other species are taken from Hann (2016). The parameters are stored internally in a list (mdib) at the index appropriate for their tree eater species code (tesp). See gendib for a full description of the equations used.

Green Weight Parameters (Table A.7)

Source: Queried FIA species reference table

Stored externally in: gwtparms.txt

Stored internally in: gwp [sp]

Table A.7: Green Weight Parameter Fields

Field	Units	Index	Format	Description
BARK_VOL_PCT	%	0	Float	The volume of bark as a percent of wood volume
WOOD_SPGR	NA	1	Float	The specific gravity of wood
BARK_SPGR	NA	2	Float	The specific gravity of bark
MC_WOOD	%	3	Float	The default moisture content of wood
MC_BARK	%	4	Float	The default moisture content of bark

Green weight parameters are taken directly from the FIA species reference table and used to convert our harvest volumes into green ton weights for transport. Both the wood and bark specific gravities use a reference point of 62.4 pounds per cubic foot and both wood and bark moisture contents are measured on a dry weight basis. Green weight parameters are stored in a dictionary (gwp) keyed by FIA species code (sp). See calcbarkwt and calcwoodwt for a full description of the equations used.

Mortality Parameters

Source: Jain et al (2017)

Stored externally in: mrtparms.txt

Stored internally in: mrt [sp]

The mortality parameters are used for calculating the predicted volume mortality in the event of a fire as a function of tree size and species. Mortality parameters are stored in a dictionary (mrt) keyed by FVS species code (sp) with the predicted mortality proportion for trees 1-4.9” in DBH at [0], 5-9.9” at [1], 10-14.9” at [2], 15-20.9” at [3], 21-29.9” at [4], 30-39.9” at [5], and 40”+ at [6].

Facility Capability Data (Table A.8)

Source: See below

Stored externally in: fcap.txt

Stored internally in: NA

The facility capability data is a list of existing mills and potential biochar processing facilities developed in collaboration with William Hollamon and David Smith of OSU. This data is used in the model to build species accept lists. Feedstock will always be a “P” or an “S”, with the latter indicating a facility that accepts saw-quality material and the former indicating one that accepts biochar feedstock. The binary variables at [4] to [9] have a one if the mill accepts that species / species group and a zero if they do not.

Table A.8: Facility Capability Data Fields

Field	Units	Index	Format	Description
psite_id	NA	0	Int	Unique facility identifier
Name	NA	1	String	Facility name
Products	NA	2	String	Facility products
Feedstock	NA	3	String	Facility feedstock
PSME	NA	4	Int	Binary variable determining if facility accepts PSME
CADE	NA	5	Int	Binary variable determining if facility accepts CADE
PIPO	NA	6	Int	Binary variable determining if facility accepts PIPO
PILA	NA	7	Int	Binary variable determining if facility accepts PILA
PICO	NA	8	Int	Binary variable determining if facility accepts PICO
TFIR	NA	9	Int	Binary variable determining if facility accepts TFIR
City	NA	10	String	City associated with facility
State	NA	11	String	State associated with facility

Travel Cost Data

Source: See below

Stored externally in: tcost.txt

Stored internally in: travel [standid] [material] [fid]

The travel cost data is the round trip cost per green ton for the optimal route from each stand (standid) to each facility (fid). This cost data was calculated using the network analyst tool in ArcMap 10.4.1, the USGS National Transportation Datasets for Oregon and California, and truck capacity and prices provided by John Sessions of OSU. Travel cost data is stored in a dictionary keyed by standid, the material being moved (with a zero for biochar feedstock and a one for saw quality material), and the fid.

Yarding Data

Source: Queried BIOSUM master database

Stored externally in: yard.txt

Stored internally in: yard [standid]

The yarding data is the percent slope and average yarding distance in feet for each stand (standid) in the project area. The slope is taken directly from FIA measurements and the average yarding distance is calculated as the distance from fuzzed plot location to the nearest road. Yarding data is stored in a dictionary keyed by standid.

Species Accept Lists

Source: Generated during Dictionary Loading

Stored in: mcap [fid]

Species accept lists store what material and species each facility accepts. Material is stored at as a zero or one at [0], with a zero indicating that the facility accepts biochar feedstock and a one indicating that the facility accepts saw-quality material. Indexes [1] through [6] indicate whether or not the facility accepts a given species / species group, with PSME at [1], CADE at [2], PIPO at [3], PILA at [4], PICO at [5], and TFIR at [6]. All species accept lists are stored in a dictionary keyed by facility ID (fid).

Optimal Route Costs

Source: Generated by evaltravel

Stored in: tcost [standid]

Optimal route costs store the cost to send material from a given stand to the lowest cost facility capable of handling that material, with the cost for biochar feedstock at [0], saw-quality PSME at [1], saw-quality CADE at [2], saw-quality PIPO at [3], saw-quality PILA at [4], saw quality PICO at [5], and saw quality TFIR at [6]. All optimal route costs are stored in a route cost dictionary (tcost) keyed by stand (standid).

Cut Trees (Table A.9)

Source: Generated by incut

Stored in: trx [rx] [standid] [treeid]

Table A.9: Cut Tree Fields

Field	Units	Index	Format	Description
Species	NA	0	Int	FVS Species code
DBH	in	1	Float	Diameter at breast height in inches
Height	ft	2	Float	Height of the tree in feet (estimated for broken tops)
TrunHt	ft	3	Float	Truncated height of a tree with a broken top, in feet
PctCr	%	4	Float	Percent crown ratio
Mdefect	%	5	Float	Percent defect in merchantable volume
TPA	NA	6	Float	Trees per acre associated with the cut tree record
MCuFt	ft3	7	Float	Merchantable volume in cubic feet
StandID	NA	8	String	Stand identifier associated with the tree
TESC	NA	9	Int	Tree Eater species code
TreeID	NA	10	Int	Tree identifier
HCB	ft	11	Float	Height to crown base in feet
YearCut	NA	12	Int	Model year the tree was cut
Rx	NA	13	Int	Prescription associated with the cut tree record
Tprice	$	14	Float	The total price of the tree in dollars
Spulp	ft^3	15	Float	The volume of rejected material in cubic feet
LogPulp	ft^3	16	Float	The volume of biochar feedstock in cubic feet
MerchVol	ft^3	17	Float	The volume of saw-quality material in cubic feet
LogPulpGT	gt	18	Float	The green weight of biochar feedstock in tons
MerchGT	gt	19	Float	The green weight of saw-quality material in tons

Cut trees are used to store all the data associated with trees harvested during a fuels treatment. All values are calculated for a single tree and stored in a dictionary (trx) keyed by prescription (rx), stand (standid) and tree (treeid). See evaltree and the associated sub-functions for a full description of how individual cut tree values are calculated.

DIB Lists

Source: Generated by dibtree

Stored in: NA

When cut trees are being bucked the model creates a DIB list for the tree being processed, an ordered list containing the diameter inside bark at one foot increments. The model assumes a one foot stump and a minimum log length of eight feet, so the first nine indexes ([0] through [8]) of each dib list are left blank to save on processing time. By default, DIB lists are not stored after the tree in question has been bucked.

Node Lists

Source: Generated by gennodes

Stored in: NA

Like DIB lists, node lists are created while a cut tree is being bucked. Where dib lists store the diameter inside bark at a given height, node lists store the value of the optimal path to a given height. Upon creation, the first nine entries of a node list are empty, but once bucking is complete the best total value of the tree is saved to [0] for easy retrieval. By default, node lists are not stored after the tree in question has been bucked.

Per Acre Outputs (Table A.10)

Source: Generated by incut

Stored in: orx [rx] [standid]

Table A.10: Per Acre Output Fields

Field	Units	Index	Format	Description
Rx	NA	0	Int	Prescription associated with the outputs
StandID	NA	1	String	Stand identifier
Tprice	$	2	Float	Gross revenue in dollars
MerchVol	ft^3	3	Float	Total saw-quality material produced in cubic feet
LogPulp	ft^3	4	Float	Total biochar feedstock produced in cubic feet
MerchGT	gt	5	Float	Total weight of saw-quality material in green tons
PulpGT	gt	6	Float	Total weight of biochar feedstock in green tons

Per acre outputs store all the yields garnered from treating a single acre of a given stand (standid) with a given prescription (rx). All per acre outputs are stored in a dictionary (orx) keyed by rx and standid.

Leave Trees (Table A.11)

Source: Generated by intree

Stored in: lrx [rx] [standid] [treeid] [year]

Leave trees store information of interest for tree records that are not wholly cut during treatment. Remaining trees per acre is calculated after treatment for each year of interest (years one, six, eleven, sixteen, and twenty) as the FVS tree list TPA minus the FVS cut list TPA (if any). Basal area is calculated for a single tree. Leave trees are stored in a dictionary (lrx) keyed by prescription (rx), stand (standid), tree (treeid), and year of record (year).

Table A.11: Leave Tree Fields

Field	Units	Index	Format	Description
Species	NA	0	Int	FVS species code
DBH	in	1	Float	Diameter at breast height in inches
TPA	NA	2	Float	Remaining trees per acre
StandID	NA	3	Str	Stand identifier associated with the tree record
TreeID	NA	4	Int	Tree identifier
Year	NA	5	Int	FVS model year
MCuFt	ft^3	6	Float	Merchantable volume in cubic feet
BA	ft^2	7	Int	Basal area in square feet
Rx	NA	8	Int	Prescription associated with tree record

Per Acre Leave (Table A.12)

Source: Generated by intree

Stored in: srx [rx] [standid] [year]

Per acre leave stores the information about what is left behind on one acre of a given stand with a given prescription. Per acre leaves are calculated for each stand for all years of interest (years one, six, eleven, sixteen, and twenty) and used primarily for calculating composite resistance scores. It’s important to note that the CBH field is only the height to crown base for single strata stands, for multi-strata stands it is the distance in feet from the top of the lowest stratum to the bottom of the next to lowest stratum. Per acre leaves are stored in a dictionary (srx) keyed by prescription (rx), stand (standid) and model year (year).

Table A.12 Per Acre Leave Fields

Field	Units	Index	Format	Description
Rx	NA	0	Int	Prescription associated with the record
StandID	NA	1	Str	Stand identifier associated with the record
Year	NA	2	Int	FVS model year
TBA	ft^2	3	Float	Basal area in square feet
RBA	ft^2	4	Float	Basal area of fire resistant species in square feet
CBH	ft	5	Float	Height to crown base / crown connectivity in feet
CBD	kg/m^3	6	Float	Crown bulk density in kilograms per cubic meter
TVOL	ft^3	7	Float	Total volume in cubic feet
PMV	ft^3	8	Float	Predicted volume mortality in cubic feet

Per Acre Costs (Table A.13)

Source: Generated by costst

Stored in: crx [rx] [standid]

Per acre costs store the costs incurred by assigning one acre of a given stand to a given prescription. Forwarder, harvester, and loading costs are all calculated using the cost data and productivity models created from Pilot Project observations. Transport costs are calculated as the sum of costs incurred in sending all biochar feedstock and saw-quality material to the lowest cost facility appropriate for that material.

The alt field is a binary variable used to generate warnings when no transportable saw-quality or biochar feedstock is produced by a treatment, with zero indicating that volume was produced for transport and a one indicating that no transportable volume was produced. Due to productivity model design Rx Evaluator may significantly underestimate costs in scenarios where no transportable material is produced.

Per acre costs are stored in a dictionary (crx) keyed by prescription (rx) and stand (standid).

Table A.13: Per Acre Cost Fields

Field	Units	Index	Format	Description
Rx	NA	0	Int	Prescription associated with the record
StandID	NA	1	Str	Stand identifier associated with the record
Tcost	$	2	Float	The total cost of treatment
Fcost	$	3	Float	Forwarder costs incurred during treatment in dollars
Hcost	$	4	Float	Harvester costs incurred during treatment in dollars
Lcost	$	5	Float	Loading costs incurred during treatment in dollars
Alt	NA	6	Int	See below
Mcost	$	7	Float	Transport costs incurred during treatment in dollars

Rx Outputs (Table A.14)

Source: Generated by scorerx

Stored in: optout [rx] [standid]

Rx Outputs are the final product of Rx Evaluator, and the information that is imported by Neo-Processor for optimization. The Mfac and Pfac fields are vestigial from an earlier version of the software and should always return “999”. Rx Outputs are stored in a dictionary (optout) keyed by prescription (rx) and stand (standid) and printed out to rxoutputs.txt.

Table A.14: Rx Output Fields

Field	Units	Index	Format	Description
Rx	NA	0	Int	Prescription associated with the record
StandID	NA	1	Str	Stand identifier associated with the record
Grev	$/ac	2	Float	The per acre gross revenue in dollars
Cost	$/ac	3	Float	The per acre treatment cost in dollars
MerchT	gt/ac	4	Float	The per acre green ton weight of saw-quality material
PulpT	gt/ac	5	Float	The per acre green ton weight of biochar feedstock
CRS1	NA	6	Int	The composite resistance score for year one
CRS6	NA	7	Int	The composite resistance score for year six
CRS11	NA	8	Int	The composite resistance score for year eleven
CRS16	NA	9	Int	The composite resistance score for year sixteen
CRS20	NA	10	Int	The composite resistance score for year twenty
ftc	$/ac	11	Float	The per acre cost to transport feedstock in dollars
flc	$/ac	12	Float	The per acre cost to load feedstock onto a truck

Function List

Parameter Loading

While not a true function, parameter loading is a critical step in how the model operates. When the Rx Evaluator is run the first thing that happens is that it populates key dictionaries with internal and external data. The parameters loaded in order are:

- Scribner values, from scribtable.txt
- Log prices, by species and top diameter, stored internally

- Species equivalence, determining which FVS species codes correspond to which Tree Eater species code, stored internally
- Parameters for calculating diameter inside bark, from dibparms.txt
- Parameters for calculating green weight, from gwtparms.txt
- Parameters for calculating predicted volume mortality, from mrtparms.txt
- Facility capability data, from fcap.txt
- Travel cost data, from tcost.txt, and
- Yarding data (distances), from yard.txt.

disroute

The disroute function is called any time when the model is run with one or more values in the Facility Disable List (fdisl). For each facility in the disable list it cycles through the travel cost dictionary (travel) and sets the route cost from each stand to that facility to $999.90 per green ton. As noted in the Facility Disable List description, this does not truly remove a facility from consideration and could result in anomalous outputs if all the facilities handling a given species are turned off at once.

The function modifies travel costs in place, and does not return an object.

evaltravel

Sub-functions: minroute

The evaltravel function creates and populates a route cost dictionary (tcost) with optimal route costs appropriate for each stand. It does this by cycling through each stand in the travel cost dictionary (travel) and determining the minimum cost facility for biochar feedstock and each saw-quality species / species group in that stand. The function returns the completed route cost dictionary keyed by stand (st).

minroute

The minroute function takes a stand of interest (st), the facility list (facl), travel cost data for the stand of interest, the species accept list dictionary (mcap), and a species of interest (sp) to find the lowest cost route from that stand to a facility capable of accepting the species of interest. The function returns the cost of that route as min.

evaltree

Sub-functions: gendib, dibtree, pricelog, gennodes, evalnode

The evaltree function is the core of the Tree Eater section of Rx Evaluator, finding the optimal combination of saw logs to maximize harvest value for a given tree. It does this by:

- Generating a DIB list for the tree with dibtree
- Generating an empty node list for the tree with gennodes, and
- Using evalnode to evaluate each node in the node list from [1] (the one foot stump) to the last node capable of creating a log (the node eight feet below the maximum height).

The process operates as a forward reaching algorithm (Denardo 2012), working its way up each node and saving the best results found (see evalnode for a full description of evaluation methodology). The function returns the completed node list with the dollar value of the optimal path for the tree of interest at [0].

gendib

The gendib function uses a tree of interest, a height of interest, and the appropriate DIB parameters to calculate the diameter inside bark according to the taper equations from Hann (2016) and Garber and Maguire (2003). It is important to note that the Garber and Maguire equations are in metric, so the model converts the key equation inputs to centimeters (diameter) and meters (height), performs the equations, and then converts the result back to inches.

The function returns the diameter inside bark at the height of interest as dib.

dibtree

Sub-functions: gendib

The dibtree function takes a tree of interest and the appropriate DIB parameters and creates a DIB list for the tree. It does this by:

- Generating a list populated with zeros from [0] through [8]
- Using gendib to calculate the diameter inside bark nine feet up the tree, appending it to the DIB list, and
- Generating and appending new diameter inside bark values at one foot increments up the tree until either gendib returns a diameter of less than four inches or the top of the tree is reached (for broken topped trees).

The function returns the completed DIB list as diblist.

pricelog

The pricelog function takes the length, species, and top diameter inside bark of a log as uses them to price a log with those characteristics. It does this by:

- Looking up the board foot volume (bf) of a log with that length and top diameter (topdib, rounded to the nearest inch)
- Converting that volume to thousand board foot volume (mbf)
- Identifying the price bracket for the log according to topdib, and
- Multiplying the volume in thousand board feet by the price per thousand board feet for the appropriate species and price bracket.

Prices were calculated as the average of Klamath Unit pond values for the four most recent quarters available on the State of Oregon’s Open Data website (https://data.oregon.gov/Natural-Resources/Log-Prices/4v4m-wr5p), specifically: quarters three and four of 2015 and quarters one and two of 2016. The function returns the price of the log as a float (p).

gennodes

The gennodes function takes a maximum height (either the top height of a tree or the height one foot before the top diameter drops below four inches) and generates node list of that length populated with zero at every index. The function returns the node list as nodes.

evalnode

Sub-functions: pricelog

The evalnode function takes a node list and a node of interest and identifies and evaluates all potential logs that could originate from that node. For every potential log length (length) stored in the log length list (llist) the function:

- Checks that the log would not violate the maximum height of the tree (maxht), and if it does not
- Identifies the target node (tnode) as the height of the node of interest (cnode) plus the length of the log and one foot of trim
- Checks to see if the top diameter of the log is six inches or greater, and
    - If it is, the value of the log is calculated with pricelog and added to the value of the best path going through the node of interest (nodes [cnode]) as the new path value (pathval)
    - If it is not, the value of the best path going through the node of interest is brought forward as the new path value
- Lastly, the new path value is checked against the current value in the target node (nodes [tnode]), overwriting that value if the new path value is higher.

The function returns the updated node list as nodes.

evalpulp

The evalpulp function takes the node list for an evaluated tree and determines the volume and characteristics of all non-saw bole material to a four inch top. It does this by:

- Identifying the optimal value of the tree
- Finding the lowest height where that value occurs (optht)
- Calculating the distance between optht and the maximum height for the tree (maxht) as pulplen, and
- Calculating the volume between optht and the maxht as a tapered cylinder.

The function returns a list (pulp) with the volume at [0] if pulplen is at least eight feet long or at [1] if the pulplen is less than eight feet long.

calcbarkwt and calcwoodwt

The calcbarkwt and calcwoodwt functions take a volume, species code (spcd), and assumed moisture content and uses them in conjunction with green weight parameters to convert volumes to weights in green pounds. They do this using:

(1 + bmc / 100) * (bvp / (100 + bvp)) * bsg * 62.4 * volume = bark biomass in green pounds, and

(1 + wmc / 100) * (1 - (bvp / (100 + bvp)) * wsg * 62.4 * volume = wood biomass in green pounds

Where:

- bmc and wmc are the assumed moisture content for bark and wood respectively, on a dry weight basis as set by the assumed moisture content user controls
- bvp (gwp [spcd] [0]) is the bark volume percent, the volume of bark as a percent of the total wood volume, taken from the FIADB species reference table
- bsg (gwp [spcd] [2]) and wsg (gwp [spcd] [1]) are the green volume to dry weight specific gravity for bark and wood respectively, taken from the FIADB species reference table
- 62.4 is the reference density for the specific gravities (in pounds per cubic foot), and
- volume is the cubic foot volume being converted.

Both functions internally label assumed moisture content mc, but Rx Evaluator runs them using one of two different global variables (mcbark and mcwood). As mentioned in the user controls section, if either function would have an assumed moisture content of less than 10%, that function will default back to a 10% assumed moisture content. The functions return wet weight in pounds as barkwt and woodwt respectively.

incut

Sub-functions: evaltree, calcbarkwt, calcwoodwt

The incut function reads in all the external trees in the cut list for a given treatment and uses them to generate a dictionary of all cut trees for that treatment and the per acre outputs for all stands treated. It does this by loading the XXXc.txt file, reading through it line by line, and for each line:

- Generating a temporary list (tlist) of the values on that line
- Setting the truncated height (d) equal to the total height (c) if it has a value of zero
- Checking if the FVS species code (a) has a corresponding Tree Eater species code and, if not, setting the Tree Eater species code (j) to zero
- Calculating the height to crown base (l)
- Generating a partial cut tree record (tree)
- Determining if the partial cut tree record is suitable for bucking and either
    - Bucking it with evaltree and appending the outputs to the partial cut tree, or
    - Assuming the entire volume (to a four inch top) will go to biochar feedstock and appending the relevant values to the partial cut tree
- Calculating and green weight of wood and bark with calcbarkwt and calcwoodwt, converting them to tons, and appending those values to complete the cut tree record.

Once the cut tree record is complete the function creates a per acre output object for the stand the tree is in (if one doesn't already exist), and adds the per acre revenue, volumes, and weights of the cut tree to the running totals for the stand. After each tree in the cut list has been processed in this way, the function returns a tuple containing a dictionary of cut trees (trx) at [0] and a dictionary of per acre outputs (orx) at [1].

loadprx

The loadprx function reads in the Potfire values for every stand under a given prescription from the XXXp.txt file, returning those values as a dictionary (prx) keyed by stand (a) and year (b). The function does not store values for stands that are not treated under the prescription being loaded, unless that prescription is the no action alternative (code 999).

loadhrx

The loadhrx function reads in the structure class inputs for every stand under a given prescription from the XXXs.txt file, and calculating the appropriate canopy connectivity metric for that stand. For stands with one or less strata (c <= 1), this is the height to crown base (d). For two strata stands (c = 2) this is the distance between the bottom of stratum one (tlist [4]) and the top of stratum two (tlist [5]). For three strata stands (c = 3) this is the distance between the bottom of stratum two (tlist [6]) and the top of stratum three (tlist [7]).

The function returns a structure class input dictionary (hrx) keyed stand (a) and year (b). Like loadprx, loadhrx does not store values for stands that are not treated under the prescription being loaded, unless that prescription is the no action alternative (code 999).

intree

Sub-functions: calcmort

The intree function loads all the external trees for a prescription from the XXXl.txt file and uses them to generate leave tree and per acre leave records for all stands treated under the prescription (or all stands when loading the no-action alternative). It does this by reading through the file line by line and for each line:

- Checking if the stand the tree is in was treated by checking the cut tree dictionary (trx) for that stand (d)
- Checking if the year of the record (f) is one of the years of interest (years one, six, eleven, sixteen, and twenty)

- Calculating the remaining TPA (c) as the tree list TPA minus the cut list TPA (if any) for that tree
- If the tree is in a stand of interest, a year of interest, and has a non-zero remaining TPA, the basal area (h) for that tree is calculated and a leave tree record is generated (treeyear).
- If a leave tree was generated, a per acre leave is created for the stand that tree is in (if one does not already exist) and the values for that tree are added to the running totals in that per acre leave.

Once all tree records in XXXl.txt have been processed, the function returns a tuple containing the leave tree dictionary (lrx) keyed by stand (d), tree (e), and year (f) at [0] and the per acre leave dictionary (srx) keyed by stand and year at [1].

calcmort

The calcmort function is used by intree to calculate the predicted volume mortality for each leave tree record to add that value to the per acre leave. It does this by looking up the mortality parameter appropriate for that tree (by species and DBH) and multiplying the volume of the tree by that parameter. The function returns the predicted volume mortality in cubic feet as mort.

costst

The costst function takes the per acre output for a stand under a given treatment and calculates the per acre cost as a function of harvesting and transportation costs. It does this by:

- Converting key inputs to metric equivalents
- Estimating the number of bunks per acre given the per acre material loading
- Calculating the productive machine minutes required for the harvester (ht), forwarding (ft), and loading (lt) and the associated costs for each (hc, fc, and lc respectively)
- Calculating the transport cost (mc) for each cut tree, and

- Summing all costs into a per acre total and compiling all the costs into a per acre cost

The bunk count per acre is estimated by dividing the total weight of material removed per acre (in green tonnes) by twelve, and rounding the result to the nearest whole number.

Harvester time is calculated as:

$$ht = 1.43tw + 0.25hdi$$

Where ht is the harvester time required in productive machine minutes, tw is the total weight of material handled in green tonnes, and hdi is the distance traveled in meters. hdi is held constant at 531 meters per acre and was calculated as two times the linear distance required to cover one acre with a corridor fifty feet wide. This may overestimate the actual distance required on flat ground where opportunistic wandering may allow the harvester to move from section to section without backtracking. It may also underestimate the actual distance on short, steep corridors where it fails to account for travel time between corridors.

Harvester cost in dollars per acre is calculated as:

$$hc = ht / 60 * 278.95 \text{ on tethered ground, and}$$

$$hc = ht / 60 * 212.47 \text{ on untethered ground}$$

Forwarder time is calculated as:

$$ft = 0.05fdi + 2.19sac + 4.06pac \text{ on tethered ground, and}$$

$$ft = 0.01fdi + 2.19sac + 4.06pac \text{ on untethered ground}$$

Where ft is the forwarding time required in productive machine minutes, sac is the weight of sawlog material in green tonnes, pac is the weight of feedstock material in green tonnes, and fdi is the distance traveled by the forwarder in meters. fdi is calculated by multiplying the number of bunks per acre by the two times the average yarding distance. fdi is also subject to a minimum distance of 473 meters per acre, calculated as two times the linear distance required to cover one acre with a corridor sixty feet wide, with an additional thirty meters added per bunk to account for out of corridor travel to a pile.

Forwarder cost in dollars per acre is calculated as:

$$fc = ft / 60 * 198.11 \text{ on tethered ground, and}$$

$$fc = ft / 60 * 172.22 \text{ on untethered ground}$$

Loading time is calculated as:

$$lt = (sac / lsp) + (pac / lpp)$$

Where lt is the time spent loading in productive machine hours, lsp is the loading rate for saw material in green tonnes per hour and lpp is the loading rate for feedstock material in green tonnes per hour. Loading cost is then calculated as the loading time multiplied by $213.22 ($172 per productive machine hour for the forwarder plus $46 per hour for the idle truck).

Per acre transport costs (mc) are constructed by retrieving and summing the optimal route costs for each portion of each cut tree on the stand. All costs are then summed into a single total cost (tc) and used to construct a per acre cost for the stand. The function returns the per acre cost as a list.

costrx

Sub-functions: costst

The costrx function uses the costst function to generate a per acre cost for each stand treated by a given prescription. It does this by taking the per acre output dictionary for that prescription (orx) and running the costst function on every stand found in it. Once complete, the function reports the number of per acre costs generated and the number of stands that failed to produce transportable material (as having incurred “brushing costs”). If a significant number of stands are failing to produce transportable material it is recommended that those stands be removed from the analysis. This can be done before running Rx Evaluator by removing them from the stand list during the FVS runs of problem prescriptions, or in Neo-Processor by adding a modest additional cost per acre and limiting the selection criteria to self-paying prescriptions only.

The function returns a per acre cost dictionary as crx.

scorest

The scorest function takes a per acre leave (pal) and uses the information in it to generate the composite resistance score for that per acre leave. It does this by generating an empty list and appending the appropriate scores as follows:

- Crown base height / crown connectivity (pal [5])is scored by giving zero points for heights / distances of less than seven feet, one point for seven to twenty feet, two points for twenty to thirty feet, and three points for heights / distances of thirty feet or greater.
- Crown bulk density (pal [6]) is scored by giving zero points for bulk densities of greater than 0.15 kilograms per cubic meter, one point for between 0.15 and 0.10, two points for between 0.05 and 0.10, and three points for bulk densities of less than 0.05 kilograms per cubic meter.
- Resistant basal area portion (rbp) is scored by giving zero points if 25% or less of the total basal area being in resistant species, one point if between 25% and 50% is in resistant species, two points if between 50% and 75% is in resistant species, and three points if more than 75% of the total basal area is in resistant species.
- The predicted volume mortality is score by giving zero points if less than 2% of standing volume is predicted to survive a fire with six to eight foot flame lengths, one point if between 2% and 30% is predicted to survive, two points if between 30% and 60% is predicted to survive, and three points if more than 60% of standing volume is predicted to survive a fire with six to eight foot flame lengths.

Once all four categories have been scored, those scores are summed and the function returns a tuple containing the list of scores (score) at [0] and the sum of those scores (crs) at [1].

scorerx

Sub-functions: scorest

The scorerx function generates the Rx outputs for a given prescription. It does this cycling through every stand with a per acre leave in the per acre leave dictionary for that prescription (srx [rx]) and building the Rx output by:

- Generating a partial Rx output (opt) with revenues, costs, and weight of materials produced for the prescription and stand being processed
- Generating and appending the composite resistance scores (crs) for each year of interest (one, six, eleven, sixteen, and twenty) in order, and
- Appending two "999" values to the end of the Rx output.

The two 999 values are from a vestigial function in an earlier version of Rx Evaluator. The appending of composite resistance scores uses a try/except functionality to accommodate stands that are clearcut (and, by extension, have no leave tree records or per acre leaves). Stands that are clearcut are assigned a composite resistance score as though 100% of basal area was in resistant species, the crown base height was 100 feet, a crown bulk density of zero, and zero predicted volume mortality in the event of a fire.

The function returns a dictionary of Rx outputs (optout) keyed by stand.

loadblock

Sub-functions: incut, costrx, loadprx, loadhrx, intree, scorerx

The loadblock function is used to automate all the other functions in Rx Evaluator into a single, easily iterable process. Given a valid prescription code, assumed bark and wood moisture contents, and slope breakpoint, the function:

- Loads external trees from the cut list and generates cut trees and per acre outputs with incut
- Calculates per acre costs with costrx
- Loads Potfire values and structure class inputs with loadprx and loadhrx respectively
- Loads external trees from the tree list and generates leave trees and per acre leaves with intree, and
- Generates Rx outputs for the prescription with scorerx.

Like scorerx, loadblock returns a dictionary of Rx outputs as optout.

Neo-Processor

Model Flow

Where Rx Evaluator performs tree level optimization and outputs stand level data, Neo-Processor takes that stand level data and uses it to perform landscape level optimization. Like Rx Evaluator, Neo-Processor is a custom script written in Python 2.7 and executed within the Spyder integrated development environment. Neo-Processor requires the user to set values for eighteen user control variables (as described in the User Controls section) and supply two external files:

- A tab delimited text file containing the unique stand identifier and area (in acres) of each stand, named acres.txt, and
- The output file from Rx Evaluator, named rxoutputs.txt.

In the process of running, Neo-Processor passes through three distinct phases, initialization, optimization, and outputting.

Initialization begins by importing the stand level data from acres.txt and rxoutputs.txt. This data is stored internally on a stand by stand basis. Those stands are then sorted into two groups, those with at least one valid fuels treatment, and those without. To save on processing time, stands with no valid fuels treatments are categorically excluded from optimization. Within the project area 354 of 825 stands (representing 45% of total federal forested acres) are excluded in this way.

Stands within the Upper Klamath are highly heterogeneous in terms of size, ranging from 3 to 7000 acres with an average size of over 2000 acres per stand. This means that if we were to optimize the landscape on a stand by stand basis that size and heterogeneity would serve to create an unrealistically discrete solution space. To resolve this, Neo-Processor uses fragments and not whole stands as the primary unit for decision-making.

Fragment size is specified by the user prior to running Neo-Processor. Then, during initialization the acre count of each stand is assessed one by one. If the acre count of the stand is less than the fragment size, that stand is used to create a single fragment. If the acre count is larger than the fragment size a fragment of that size is created and the acre count for the stand is reduced by that number of acres. This process is iterative, repeating until the number of acres remaining is less than the fragment size, and the remaining acres made into one final fragment. To illustrate, if you took a stand of 3500 acres and a fragment size of 1000 acres, that stand would create three fragments of 1000 acres each, and one fragment of the remaining 500 acres.

It must be noted that this system has to potential to effectively price out a small number of acres in each stand. For example, if you took a stand of 3002 acres and a fragment size of 1000 acres, you would create three fragments of 1000 acres each, and one fragment of 2 acres. That two acre fragment would be unlikely to ever breakeven due to the application of fixed costs per entry. This is problematic in theory, but has virtually no effect on the quality of solutions in practice, the few dozen acres adversely affected dwarfed by the hundreds of thousands of acres under consideration.

Once all stands have been turning into fragments, Neo-Processor finds and evaluates every unique fragment that could exist. A fragment is considered unique if no other fragment shares the same parent stand, prescription, and fragment size. No stand should produce more than two unique fragments per valid prescription, with one of the fragment size, and one of less than the fragment size. For each unique fragment, the outputs for treating that fragment are calculated including:

- Gross revenue
- Treatment cost
- Sawlog material produced
- Biochar feedstock produced
- The composite resistance score for years one, six, eleven, sixteen, and twenty, and
- Any additional per acre and/or fixed costs specified by the user

Fixed costs are defined in part by an average entry size (specified by the user) and will scale up when fragments are larger than the average entry size. Fixed costs do not scale down for fragments smaller than the average entry size to better represent costs that are largely insensitive to treatment size, such as those associated with mobilization.

Once all fragments are created and evaluated, an initial solution is generated through one of two mechanisms:

- If there is no even flow constraint and no limits on the number of acres treated in any period, a fully random initial solution is generated.
- If there is an even flow constraint or a limit on the number of acres treated in any period, a partially random solution is generated.

Fully random solutions are generated by moving through each fragment in order and assigning it one of the prescriptions appropriate for that fragment's parent stand, chosen at random. Partially random solutions are generated by

- Assigning a user-defined percent of fragments (selected randomly) to a prescription appropriate for that fragment's parent stand, chosen at random. If an assignment would violate the limitation on acres treated, that fragment is assigned the no-action alternative instead.
- The remaining fragments are used to generate a randomized list. The model then
    - Finds the treatment period with the lowest yield of interest (if even flow is enabled) or the period with the lowest number of acres treated (if even flow is disabled).
    - Cycles through the list until it finds a fragment with a treatment capable of improving that minimum
    - Assigns that fragment a prescription that improves the minimum (selecting one at random if more than one treatment is capable of doing so)
    - Updates the yield of interest and/or number of acres treated, and
    - Removes the fragment from the list. If prescription assignment would violate the limit on acres treated, that fragment is ignored as though it had no valid prescriptions capable of improving the minimum.
- This process repeats until either all fragments have been assigned a treatment, or no fragments are capable of improving the minimum. When no fragments are capable of improving the minimum, all remaining fragments are assigned the no-action alternative.

Early Neo-Processor designs struggled to effectively apply even flow constraints due to the extremely lop-sided nature of the solution space. Many stands are inoperable in early periods, but grow into operability over the course of the planning horizon. This operability in-growth combined with the fact that late-period prescriptions are removing more material results in fully random solutions that are highly skewed towards later periods in terms of acres treated and volumes produced.

Trial runs with fully random solutions and even flow constraints often spent half or more of their total move count simply seeking feasibility, resulting in very poor quality solutions. By contrast, partially random solutions virtually always start within acceptable bounds, resulting in much higher quality solutions.

With the outputs for each fragment calculated and the initial solution generated, the model begins the actual work of optimization. This process can be broken into three distinct phases: move generation, move evaluation, and move implementation.

During move generation a random fragment is selected, the model looks up and copies the list of all prescriptions that could be applied to that fragment, removes the current prescription from the list, and selects one of the remaining prescriptions at random. Once both prescriptions are identified, a move effect is created. The move effect is an ordered list containing the revenue, cost, and biomass produced by both treatments, and the net effect on composite resistance score of moving from the old treatment to the new treatment in years one, six, eleven, sixteen, and twenty.

Move evaluation analyzes that move effect list. In order, it tests the move effect for:

- Violations of the limits on acres treated per period
- Violations of the even flow constraint, and
- The quality of the objective function.

Moves that would cause a violation in the number of acres treated are always rejected. Moves that would cause a violation in the even flow constraint are rejected unless the new solution is considered closer to feasibility than the current solution. Even flow is measured within the model as the percent of deviation in any period from the average for all periods in a given yield of interest (such as net revenue, acres treated, or biochar feedstock produced). For example, if the average net revenue for all periods was $10,000 and the even flow constraint was set to 30% on net revenue, any move that resulted in a period with a net revenue higher than $13,000 or less than $7,000 would be rejected as infeasible unless the current solution was also infeasible. When both the current and new solution are infeasible, the squared deviation from the average is measured for each period and summed for both solutions. Then, the new solution will be accepted if it has a lower sum of squared deviations than the current solution.

The quality of the objective function is tested using a modified Great Deluge algorithm (Dueck 1993). Moves that do not cause a violation and are an improvement on the objective function are automatically accepted. Moves that do not cause a violation and reduce the objective function may also be accepted as an allowable disimprovement. What constitutes an allowable disimprovement changes over the course of each run according to the following rules:

- An allowable disimprovement may be no worse than the current objective function minus the “flood” value.
- The flood value is set to zero during solution generation and defaults back to zero if it would ever become negative.
- If the flood value is zero and an improvement is found, the flood value is set to 99% of the value of that improvement.
- If the flood value is a non-zero number, an improvement is found, and the improvement is less than or equal to ten times the current flood value, the current flood value is reduced by 50% of the value of the improvement. Lastly,
- If the flood value is a non-zero number, an improvement is found, and that improvement is more than ten times the current flood value, the flood value is set to 99% of the value of that improvement.

This resetting flood system is designed to force the model to become less and less accepting of disimprovement when repeatedly making small gains and disallow backtracking (where the model cycles back and forth between functionally identical moves), but still allow it to effectively explore the solution space when making gains in leaps and bounds.

Neo-Processor’s move evaluation also includes a “noise generation” function, designed to prevent the model from becoming trapped in local maxima for extended periods of time when limitations on the number of acres treated are enabled. If enough moves are rejected for violating the limitation on acres treated, noise generation is triggered. When noise generation is triggered, fragments are randomly selected and set to the no-action alternative until 10% of all fragments have been reset in this way.

When a move is accepted, it is immediately implemented. The fragment subject to the move is updated to the new prescription and the running totals for all yields of interest, the objective function, and the number of acres treated are all updated using the information from the move effect list. If the objective function value is better than the best objective function value ever seen or the solution is closer to feasibility than any solution ever seen, that solution is saved away for future use.

Neo-Processor attempts a set number of moves in each run. The total number of moves is determined by the user as a function of the number of fragments being analyzed. The number of moves required to find a high quality solution varies by fragment size and the nature of the constraints applied, with more limiting constraints generally requiring more moves per fragment.

After running through the entire move count for a given run, Neo-Processor loads the best solution ever seen and reports the following:

- The total number of valid stand / prescription combinations
- The total number of acres in the project area, the number of acres broken into fragments, and the number of acres represented by all fragments
- The number of unique fragments identified and evaluated
- The type of initial solution generated (fully random or partially random)
- The final flood value
- The best objective function value ever found
- The number of improvements, accepted disimprovements, and rejected disimprovements
- The number of moves rejected due to infeasibility and the number of times noise generation was triggered (if any)
- The number of acres assigned to each treatment by the best solution ever found
- The number of acres treated in each period by the best solution ever found
- The net revenue, gross revenue, sawlog material, and biochar feedstock produced in each period by the best solution ever found
- The costs incurred in each period by the best solution ever found

- The composite resistance scores for years one, six, eleven, sixteen, and twenty for the entire project area under the best solution ever found and the no-action alternative, and
- The composite resistance scores for years one, six, eleven, sixteen, and twenty for operable acres only (those with a valid fuels treatment) under the best solution ever found and the no-action alternative.

In addition, the model outputs two tab delimited text files. One contains a list of all fragments with their parent stand identifier, acre count, and prescription under the best solution ever found. The other reports the total weight of material delivered to each facility in each period.

User Controls

Neo-Processor includes eighteen user control variables meant to allow the user to quickly and easily fine tune model runs without significantly altering the program itself. These variables are found in a single code block from line 957 to line 974.

Initial Solution Control
Variable name: init
Default value: 0
Appropriate inputs: 0, 1, 2, 3, 999

The initial solution control serves two purposes: determining how initial solutions are generated with gensol, and determining whether or not the model disables prescriptions in the disable list (disl).

- If init is set to 0, the solution is randomly generated from all potential prescriptions
- If init is set to 1, the solution is randomly generated from all potential prescriptions except those in the disabled list
- If init is set to 2, the solution is randomly generated from all self-paying prescriptions

- If init is set to 3, the solution is randomly generated from all self-paying prescriptions save those in the disabled list
- If init is set to 999, each fragment is automatically assigned the no action alternative. This is generally only used for performing diagnostics and isolating errors.

While genlimsol does not directly use the initial solution control, it must still be set to one or three to disable prescriptions when using limited solutions.

Disable List

Variable name: disl

Default value: []

Appropriate inputs: any valid treatment code or combination of treatment codes

The disable list allows the user to remove treatments from consideration by entering any number of treatment codes in the list to disable them. Treatment codes must be entered between the square brackets with a comma separating each one. Note: this does not remove the treatments from Rx Outputs or Fragment Outputs, it only prevents the model from selecting the disabled prescriptions during initial solution generation and move generation.

Moves per Fragment Control

Variable name: mpf

Default value: 1000

Appropriate inputs: any positive integer

The moves per fragment control works with the fragment size control to determine how many total moves the model makes before reporting the best solution found. There is no effective limit on the number of moves the model can make, but higher move counts will increase running time. Moves per fragment should be modified in tandem with fragment size.

Fragment Size Control

Variable name: fragsize

Default value: 1000

Appropriate inputs: any non-negative number

The fragment size control determines the maximum fragment size in acres. Smaller fragment sizes allow for a more granular solution space when using even flow and/or area control constraints, but will increase running time. Moves per fragment and fragment size should be modified in tandem.

Biochar Feedstock Price Control

Variable name: biop

Default value: 0.0

Appropriate inputs: any non-negative number

The biochar feedstock price control determines the price in dollars per green ton associated with the sale of biochar feedstock.

Additional Variable Cost Control

Variable name: adac

Default value: 0.0

Appropriate inputs: any non-negative number

The additional variable cost control is used to apply additional dollar per acre costs to each fuels treatment in the analysis. This can be used to account for site prep, piling, burning, or any other cost that can be reasonably expressed on a per acre basis.

Entry Size Control

Variable name: ents

Default value: 100.0

Appropriate inputs: any non-negative number

The entry size control determines the assumed average entry size (in acres) for use in calculating fixed treatment costs.

Fixed Cost Control

Variable name: fixc

Default value: NA

Appropriate inputs: any non-negative number

The fixed cost control is used to apply a fixed dollars per entry cost to each fuels treatment. This includes but is not limited to mobilization costs. Fixed costs scale up if the fragment size is larger than the entry size, but do not scale down for fragments smaller than the entry size.

Objective Function Control

Variable name: obj

Default value: NA

Appropriate inputs: 0, 1, 2, 3

The objective function control determines what is being maximized for the current model run. It has four built in settings:

0 – Maximizing total net revenue over the planning horizon.

1 – Maximizing total gross revenue over the planning horizon.

2 – Maximizing total biomass removed over the planning horizon.

3 – Maximizing total composite resistance score over the planning horizon.

Self-Paying Mode Control

Variable name: spmode

Default value: 0

Appropriate inputs: 0, 1

The self-paying mode control determines whether or not the model can select revenue negative treatments when seeking a solution. Setting spmode to zero allows all treatments, setting it to one allows self-paying treatments only.

Even Flow Mode Control
Variable name: emode
Default value: 0
Appropriate inputs: 0, 1, 2, 3, 4, 5

The even flow mode control determines whether or not the even flow constraint is enforced and what it is enforced on. Neo-Processor has six built in settings:

0 – Even flow constraint is disabled.

1 – Even flow constraint is applied to net revenue.

2 – Even flow constraint is applied to gross revenue.

3 – Even flow constraint is applied to sawlog volume.

4 – Even flow constraint is applied to biochar feedstock volume.

5 – Even flow constraint is applied total acres treated.

Even Flow Control
Variable name: even
Default value: 0
Appropriate inputs: any positive integer between one and one hundred

Where even flow mode determines whether or not the even flow constraint is enforced, the even flow control determines how limiting that constraint is. When even flow is enabled, moves are rejected as infeasible if they would result in the yield at the period of interest being more than even percent away from the average yield for all periods, unless that move would result in a solution with less total deviation from the average than the current solution.

Acre Limitation Mode Control

Variable name: almode

Default value: 0

Appropriate inputs: 0, 1

The acre limitation mode control determines whether or not a limit is placed on the maximum number of acres treated per planning period. Setting almode to zero allows any number of acres to be treated in any period, while setting it to one enforces the limits as defined in the actlim variable.

Acre Limitation Control

Variable name: actlim

Default value: [“l”, -1, -1, -1, -1]

Appropriate inputs: -1, 0, or any positive number

Just as even determines how limiting the even flow constraint is, actlim determines how limiting the acre limitation constraint is. The control itself is an ordered list (see results in the object list for a full description) that allows the user to determine how many acres may be treated by the model in any given period. For example, to limit the model to 50,000 acres in the first period and 75,000 in each other period, you would enter [“l”, 50000, 75000, 75000, 75000]. If a value of -1 is entered in each field, the model will allow any number of acres to be treated even if almode is set to one.

Random Fraction Control

Variable name: ranf

Default value: 50

Appropriate inputs: any positive integer between 0 and 100

The random fraction control is used when generating the initial solution on runs subject to even flow or acre limitation constraints. It does this by determining what percent of fragments are assigned a prescription randomly and, by extension, what percent are “guided” (see genlimsol, randomlim, and guidedlim for a full description). The default value of 50 will work for most model runs, but the more limiting your constraints the lower the random fraction should be to ensure the highest quality initial solution.

Noise Control
Variable name: noise
Default value: 10.0
Appropriate inputs: any positive number between 1 and 100

The noise control determines what percent of fragments are set to the no-action alternative when the gennoise function is triggered. Noise generation is only used when the acre limitation constraint is enabled, and is designed to prevent the model from becoming trapped in local maxima for extended periods of time.

Noise Trigger Control
Variable name: noiset
Default value: 2000
Appropriate inputs: any positive integer

The noise trigger control determines how many moves must be rejected due to the acre limitation constraint before noise generation is triggered. This number is highly sensitive to the granularity of the solution space, with more granular solution spaces benefitting from high trigger values (to fully explore the local maxima before noise generation) and less granular solution spaces benefitting from lower trigger values (to spend less total time at any given local maxima and more time finding different local maxima). It is highly recommended to test a wide variety of noise trigger values for any given solution space.

Object List

Rx Output

Source: Imported from Rx Evaluator

Stored In: rxe [standid] [rx]

Rx Outputs are ordered lists describing the per acre outputs of each valid stand / treatment combination. Rx Outputs are imported unaltered from Rx Evaluator and used only to calculate Fragment Outputs. Rx Outputs are stored in nested dictionaries keyed by stand identifier [standid] and prescription [rx].

Fragment (Table A.15)

Source: Generated by genfrags

Stored in: frags [fid]

Table A.15 Fragment Fields

Field	Units	Index	Format	Description
FID	NA	0	Int	Fragment identifier
StandID	NA	1	Str	Stand identifier
Acres	ac	2	Float	Acres in fragment
Rx	NA	3	Int	Current prescription assigned to fragment

Fragments are ordered lists that serve as the primary decision-making unit for Neo-Processor. Maximum fragment size is set by the user prior to each run with smaller fragments providing a more granular solution space at the cost of an increase in time to solve.

Fragment Output (Table A.16)

Source: Calculated by evalfrags

Stored in: rxf [standid] [rx] [ac]

Table A.16: Fragment Output Fields

Field	Units	Index	Format	Description
Rx	NA	0	Int	Treatment code
StandID	NA	1	Str	Stand identifier
Grev	$	2	Float	Gross revenue from treating fragment
Cost	$	3	Float	Cost of treating fragment
MerchT	gt	4	Float	Green tons of sawlog quality material
PulpT	gt	5	Float	Green tons of biochar feedstock
CRS1	NA	6	Int	Composite resistance score for year 1
CRS6	NA	7	Int	Composite resistance score for year 6
CRS11	NA	8	Int	Composite resistance score for year 11
CRS16	NA	9	Int	Composite resistance score for year 16
CRS20	NA	10	Int	Composite resistance score for year 20
Mfac	NA	11	Int	Lowest cost sawlog facility
Pfac	NA	12	Int	Lowest cost biochar facility
AddC	$	13	Float	User specified per acre and per entry costs

Fragment outputs are ordered lists describing the per fragment yields for each unique stand / treatment / fragment size combination. Fragment Outputs are used by multiple functions to evaluate a solution or the effects of move.

Move Effect (Table A.17)

Source: generated by genmove

Stored in: mef

Table A.17: Move Effect Fields

Field	Units	Index	Format	Description
NRx	NA	0	Int	The new prescription for the fragment
StandID	NA	1	Str	The stand identifier for the fragment being moved
FID	NA	2	Int	The fragment identifier for the fragment being moved
Ope	NA	3	Int	The current treatment period for the fragment
OG	$	4	Float	The revenue generated by the current treatment
OC	$	5	Float	The costs incurred by the current treatment
OM	gt	6	Float	The sawlog material produced by the current treatment
OP	gt	7	Float	The biochar feedstock produced by the current treatment
Npe	NA	8	Int	The new treatment period
NG	$	9	Float	The revenue generated by the new treatment
NC	$	10	Float	The costs incurred by the new treatment
NM	gt	11	Float	The sawlog material produced by the new treatment
NP	gt	12	Float	The biochar feedstock produced by the new treatment
se1	NA	13	Float	The effect of moving treatments on year one CRS
se6	NA	14	Float	The effect of moving treatments on year six CRS
se11	NA	15	Float	The effect of moving treatments on year eleven CRS
se16	NA	16	Float	The effect of moving treatments on year sixteen CRS
se20	NA	17	Float	The effect of moving treatments on year twenty CRS

A move effect is an ordered list containing all the information necessary to evaluate and make a move. When moving to or from the no action alternative the appropriate period of treatment is set to zero. The model generates and stores move effects one at a time, saving each to the mef variable.

Result (Table A.18)

Source: see below

Stored in: see below

Table A.18: Result Fields

Per Period Result	Stored As	Identifier	Calculated By	Updated By
Total CRS	cfs	NA	calccfs	updatecfs
Operable CRS	ocfs	NA	calcopcfs	NA
Gross revenue	grev	"r"	calcecn	updateyield
Costs	cost	"c"	calcecn	updateyield
Sawlog material	mton	"m"	calcton	updateyield
Biochar feedstock	pton	"p"	calcton	updateyield
Net revenue	nrev	"n"	calcnrv	updateyield
Acres treated	act	"a"	calcact	updateact
Acre limits	actlim	“l”	NA	NA
Delivered material	deld	“dX”	calcdelivered	NA

Results are ordered lists containing the per-period yields for the current solution, with a unique identifier at [0] and the yield for each period at indexes [1] through [4]. Composite resistance score results do not contain an identifier, instead holding the CRS for each year of interest in order. The total CRS result holds the average CRS for all acres in the project area, while the operable CRS result holds the average CRS for acres with a valid fuel reduction treatment.

The actlim result is a special user-defined result that prevents the model from harvesting more than the number of acres specified for each period.

Unlike the other results, the delivered material result is a dictionary of results keyed by facility number. Delivered materials are only calculated for the best solution ever found as part of the end of run analysis. Each delivered material result is assigned a unique identifier of “dX” where X is the appropriate facility number.

Solution
Source: generated by savesol
Stored in: bsol

Solutions are ordered lists storing the prescription for each fragment at the index corresponding to its fragment identifier (i.e., the prescription for fragment 0 is stored at [0], fragment 1 at [1] and so on). The model only generates and saves a solution object when a new best solution is found, saving it as bsol.

Function List

Output Loading

While not truly a compartmentalized function, output loading is the first thing Neo-Processor does and is required for the rest of the program to operate. Output Loading starts by creating a host of dictionaries and lists:

rxe – the dictionary used to store Rx Outputs.

rxl – a list of all treatment codes in Rx Outputs.

fcl – a list of all facility codes in Rx Outputs.

etd – a dictionary of all possible treatments for each stand, keyed by the stand identifier.

spd – a dictionary of all self-paying treatments for each stand, keyed by stand identifier.

per – a dictionary of all treatments in Rx Outputs, keyed by treatment period.

Once those are created, output loading opens the text file containing the Rx Outputs from RxEvaluator and for each line in the file:

- Converts the line into a temporary list (tlist)
- Saves the prescription as rx
- Adds the prescription to rxl if it is not present
- Adds the prescription to per if it not present and is not the no action alternative
- Saves the stand identifier as st
- Adds the stand identifier to rxe and etd if it is not present
- Adds the prescription to etd if it is not present
- Saves the per acre revenue and per acre costs as c and d respectively
- Adds the prescription to spd if it is not present and per acre revenues are greater than or equal to per acre costs.
- Saves the remaining Rx Output values to variables e through m
- Adds the sawlog facility and biochar facility to the facility list if they are not present
- Lastly, it combines rx, st, and variables c through m into a list and saves them to rxe [st] [rx].

After every line in the text file has been processed, Neo-Processor prints the number of stands and stand / prescription combinations encountered.

genfrags

The genfrags function is used to import the area of each stand and split each stand into fragments where appropriate. It begins by generating a set of variables:

frags – the dictionary used to store fragments

nas – a dictionary used to store stands with no valid fuels reduction treatments

frl – a list of all fragments

fcount – the current number of fragments processed

It then opens a text file containing the stand identifier and acre count for each stand on separate lines. For each line in the file it

- Converts the line to a temporary list (tlist)
- Saves the stand identifier as st
- Calculates the number of possible treatments for the stand (net) and saves the total number of acres as the current number of acres (cac)
- If the stand only has one possible treatment (the no action alternative) the stand is added to the no-action dictionary (nas) and not broken into fragments.
- If the stand has more than one possible treatment and the current acre count is greater than the maximum fragment size (fragsize), a fragment of the maximum size is generated and saved to the fragment dictionary (frags) and fragment list (frl) with the current fragment count (fcount) as its unique identifier. Then the current acre count is reduced and the fragment count increased.
- This process repeats until the current acre count is less than the maximum fragment size, and the remaining acres are used to generate one last fragment.

Once each line in the file has been processed Neo-Processor prints the total number of acres loaded, the number of acres with more than one prescription, and the number of acres with only one prescription (the no action alternative). The function returns a list containing the fragment dictionary at [0], the no action dictionary at [1], the fragment list at [2], the fragment count at [3], the number of operable acres (oac) at [4], the number of inoperable acres (nac) at [5], and the total number of acres (tac) at [6].

evalfrags

The evalfrags function identifies all potential treatments for each unique fragment and calculates the fragment outputs for each of them. It does this by creating an empty fragment output dictionary (rxf) and cycling through each fragment in the fragment dictionary (frags). For each fragment it looks up all potential prescriptions in the potential prescriptions dictionary (etd). For each prescription found for each fragment:

- If the fragment's stand identifier (st) is not present in the fragment output dictionary (rxf), it is added as a dictionary
- If the current prescription (rx) is not present in rxf [st], it is also added as a dictionary.
- If the fragment's acre count (ac) is not present in rxf [st] [rx],
    - The per acre yields are looked up for that stand / prescription combination and are multiplied by the acre count of the fragment
    - The composite resistance scores for that stand / prescription combination are looked up, multiplied by the acre count of the fragment, and divided by the total number of acres in the project area
    - The facility for sawlog material and biochar feedstock are identified
    - If the entry size (ents) is less than the acre count of the fragment fixed costs (fixc) are scaled to the number of entries required (ac / ents * fixc). Fixed costs are not scaled if the entry size is larger than the acre count of the fragment
    - Additional per acre costs (adac) are multiplied by acre count of the fragment and added to the fixed costs
    - The resulting yields, scores, facilities, and costs are built into a fragment output list and stored as rxf [st] [rx] [ac].

In general, the model should find two unique fragments per stand / treatment combination. The only exception to this occurs when the acre counts for one or more stands are evenly divisible by the maximum fragment size. The evalfrags function returns the populated fragment output dictionary.

pricebio

The pricebio function is called when the user specifies a non-zero price for biochar feedstock (biop). The function updates each fragment output in the fragment output dictionary (rxf) by multiplying the biochar feedstock produced by biop and adding that value to the existing gross revenue. The function modifies each fragment output in place and does not return an object.

disablerx

If any prescriptions are in the disabled list (disl) and the initial solution control (init) is set to one or three, the disablerx function is called. The function works by cycling through each stand in the potential prescription dictionary (etd) or self-paying dictionary (spd) and removing any prescriptions on the disabled list. This prevents those prescriptions from being selected during initial solution generation or move generation for the current run.

The function prints the number of prescriptions disabled. If all fuels treatments for one or more stands are disabled (leaving only the no action alternative), the function will also print the number of stands and acres that no longer have a valid treatment. The disablerx function modifies its targets in place and does not return an object.

gensol

The gensol function is used for generating fully random initial solutions suitable for runs without even flow or area control constraints. Like genlimsol, gensol does not generate a solution object, instead randomly selecting a prescription for each fragment and modifying that fragment in place. The function has several different settings controlled by the initial solution control (init):

- If init is set to 0, the solution is randomly generated from all potential prescriptions
- If init is set to 1, the solution is randomly generated from all potential prescriptions except those in the disabled list
- If init is set to 2, the solution is randomly generated from all self-paying prescriptions
- If init is set to 3, the solution is randomly generated from all self-paying prescriptions save those in the disabled list
- If init is set to 999, each fragment is automatically assigned the no action alternative. This is generally only used for performing diagnostics and isolating errors.

genlimsol

Sub-functions: randomlim, genptd, guidedlim, updatelim

The genlimsol function is used for generating initial solutions suitable for runs with even flow and/or area control constraints (referred to as “limited solutions”). Like gensol, genlimsol does not generate a solution object, instead modifying each fragment in place. Unlike gensol, this function has two distinct phases of operation:

- The random phase, during which a percent of the total solution is generated by random assignment with the randomlim function. How many fragments are randomly assigned is managed with the random fraction control (ranf). And
- The guided phase, during which the rest of the solution is generated by the guidedlim function. In the guided phase the model preferentially selects prescriptions to improve feasibility. If no feasibility improving moves can be found, the remaining fragments are set to the no-action alternative.

Throughout both phases the function uses a randomized fragment list (ranl) and assignment count (count) to ensure that each fragment is assigned a prescription. The function prints the number of attempted random assignments, the number of random assignments rejected for violating area control constraints, the number of guided assignments, and the number of fragments set to the no action alternative.

randomlim

Sub-functions: updatelim

The randomlim function is used to generate the random portion of a limited solution. It brings in the randomized fragment list (ranl) from genlimsol and generates two empty results, one for acres treated (act) and one for the result subject to the even flow constraint (eyield), if any. It then moves through the randomized fragment list assigning a random prescription to each one until it has filled the percent of total fragments set by the random fraction control.

The act and eyield results are updated with every move by the updatelim function. If a prescription assignment would violate the acre limitation constraint in any period, that prescription is instead set to the no-action alternative.

The randomlim function returns a tuple containing:

- A list with act at [0], the number of assignments made (count) at [1], and the number of assignments rejected and set to the no-action alternative (vioac) at [2]. And,
- The eyield result, or an "x" if the even flow constraint is disabled.

updatelim

The updatelim function is used to keep a running tally of results while generating a limited solution. It does this by looking up the appropriate value in the fragment output dictionary (rxf) and adding it to the appropriate period of the result being updated.

guidedlim

Sub-functions: updatelim, genptd

The guidedlim function is used to build the guided portion of a limited solution and utilizes the act, eyield, and count values returned by randomlim. It begins by slicing the random fragment list (ranl) to include only those fragments that were not assigned a prescription by randomlim, and then randomizing this truncated list. The genptd function is then called to produce a dictionary of treatments keyed by stand identifier and treatment period (ptd).

Once ptd is generated the model finds the eyield period with the lowest value (or the act period with the lowest value if the even flow constraint is disabled) and cycles through the truncated random fragment list until it finds a fragment with a valid prescription that can increase that minimum value, that prescription is assigned to that fragment, and that fragment is removed from the randomized fragment list. If that fragment is associated with more than one treatment capable of increasing the minimum value, one of those treatments is assigned at random.

If a prescription assignment would violate the acre limitation constraint, that fragment is ignored for that assignment as though it had no valid treatments. If the model cycles through the entire random fragment list and fails to find a fragment capable of improving the minimum all remaining fragments are assigned the no-action alternative.

The function returns a list with act at [0], the number of successful guided moves (gmove) at [1], the number of fragments set to the no action alternative (nmove) at [2], and the eyield (or "x") at [3].

genptd

The genptd function is used to generate a dictionary keyed by stand identifier and treatment period for use in generating limited solutions. It does this by creating an empty period of application dictionary (ptd) and cycling through every fragment in the fragment dictionary (frags). For each fragment it

- Adds the stand identifier (st) associated with that fragment to ptd if it is not present
- Looks up st in the prescription dictionary (rxd) and for each fuels treatment there it
    - Looks up the period of application for that treatment
    - Adds that period (rxp) to ptd [st] as a list if it is not present, and
    - Appends the treatment code to ptd [st] [rxp].

After each fragment has been checked, the function returns the completed dictionary.

calccfs and calcopcfs

The calccfs function is used to calculate the composite resistance score for years one, six, eleven, sixteen, and twenty for the current state of each fragment in the fragment dictionary (frags) and the stands in the no-action dictionary (nas). It does this by generating an empty result (cfs) and cycling through each no-action stand and fragment, calculating their contribution to the average composite resistance score for the entire project area in each year of interest, and adding those contributions to the running totals for each year. Once every no-action stand and fragment is accounted for, the function returns cfs.

The function can also be set evaluate the no-action alternative for the project area by setting mode to 999. The calcopcfs function operates in fundamentally the same way, but only looks at fragments, not no-action stands.

calcecn, calcton, and calcact

The calcecn function calculates the economic outputs for the current state of each fragment in the fragment dictionary (frags). It does this by generating two empty results (grev and cost) and cycling through each fragment, looking up its associated revenue and costs in the fragment output dictionary (rxf) and adding them to the running total. Once each fragment is accounted for the function returns a tuple containing grev at [0] and cost at [1].

The calcton function operates in fundamentally the same way, but tallies the green ton weight of sawlog material and biochar feedstock, returning their results as mton at [0] and pton at [1] respectively. The calcact function uses the same methods as calcton and calcecn, but returns the acres treated result (act) as a list, not as part of a tuple.

calcnrv

The calcnrv function calculates the net revenue for the current state of each fragment in the fragment dictionary (frags) by generating an empty result (nrev) and adding the revenue minus the cost (taken from grev and cost respectively) for each period. The function returns the completed result.

calccov

The calccov function calculates the current objective function value by summing the result (or results) appropriate for the current objective function. See obj in the user controls section for full description of objective function settings. Returns the summed value as cov.

genmove

The genmove function is used to generate move effects for evaluation. It does this by

- Selecting a random fragment from the fragment list (frl)

- Identifying the current prescription (crx) for that fragment
- Generating a list of valid prescriptions (prx) for that fragment, removing the current prescription, and selecting one of the remaining prescriptions at random as the new prescription (nrx)
- Generating a new move effect (mef) with the new prescription at [0], stand identifier (st) at [1], fragment identifier (fr) at [2], and old treatment period (ope) at [3]
- Appending the revenue, costs, sawlog material, and biochar feedstock associated with the old prescription to mef
- Appending the new treatment period (npe), revenue, cost, sawlog material, and biochar feedstock associated with the new prescription to mef, and
- Calculating the net effect of the new treatment on the composite resistance score for each year of interest (se1 through se5) and appending those effects.

The function returns the completed move effect as a list.

evalmove

Sub-functions: testactl, testeven, genbounds, testbounds, and calcdev

The evalmove function is the linchpin of the Neo-Processor program, determining whether moves created by genmove are accepted or rejected. The function uses a modified Great Deluge selection criteria, seeking improvement but capable of accepting limited disimprovement to better explore the solution space. The function is built around four key variables, the acceptance of the move (accept), the state of move assessment (fin), the tolerance for disimprovement (flood), and the improvement found (imp). The function works in three distinct stages:

- Stage one checks for violations of using the testactl and testeven functions. If the move is found to be in violation it will be assigned an accept value of -3 (rejected for violating the acre limitation constraint), -2 (rejected for violating the even flow constraint) or -1 (accepted for moving the model closer to meeting even flow constraints) and fin will be set to 1, passing the move through the other stages without further assessment.

- If fin is still set to 0, stage two evaluates the effect of the move on the objective function on its own merits. If the move would improve the objective function value, accept is set to 1 (accepted for improving the objective function), and the size of the improvement is saved as imp. If the effect of the move is no worse than the current value of the objective function minus the current flood value, accept is set to 2 (accepted as an allowable disimprovement). If the effect of the move is worse than the objective function minus the current flood value, accept remains 0 (move rejected).
- If an improvement is found, stage three is triggered, updating the flood value.

The flood value starts each run set to zero and defaults back to zero if it ever becomes negative. If the flood value is zero and an improvement is found, the flood value is set to 99% of that improvement. If the flood value is a non-zero number and an improvement is found, one of two things happens, either

- The flood value is reduced by 50% of the found improvement, or
- If the improvement found is more than ten times the current flood value, the flood value is set to 99% of the value of that improvement.

This design ensures that the model can adequately explore the solution space while improving in leaps and bounds, but will become more and more intolerant of disimprovement while fine tuning a solution.

The function returns a tuple containing the accept value at [0] and the flood value at [1].

testactl

The testactl function is used by the evalmove function to determine whether or not the move would cause the solution to become infeasible. It does this by adding the number of acres subject to the move (ac) to the current number of acres being treated in the new treatment period act [np] and comparing that number to the limitation imposed on that period (actlim [np]). If the move would generate a violation, accept is set to -3 to reject the move and fin is set to 1 to prevent the model from analyzing the move further.

The function returns a tuple containing accept at [0] and fin at [1].

testeven

Sub-functions, genbounds, testbounds, calcdev

The testeven function is used by the evalmove function to assess the effect of the move on the even flow constraint. Even flow in Neo-Processor is measured as the deviation of a result in any given period from the average of that result for all periods. If the move would cause the result for the old treatment period (oy) or the result for the new treatment period (ny) to be more than even percent away from the average, fin is set to zero and accept is set to -2. When this occurs the model performs a deviation test, calculating the sum of squared deviations of the new solution (ndev), comparing it to the sum of squared deviations of the old solution (odev), and setting accept to -1 if the new deviation is lower than the old.

Like testactl, testeven returns a tuple containing accept at [0] and fin at [1].

genbounds

The genbounds function finds the average (ave) for the result subject to the even flow constraint (cyield) and uses that average to calculate the minimum acceptable value for any given period (lb) and the maximum value acceptable for any given period (ub).

The function returns a tuple containing the upper bound at [0] and the lower bound at [1].

testbounds

The testbounds function compares the result in a period of interest (tp) to the upper and lower bounds calculated by genbounds. If that result of interest (cyield [tp]) is more extreme than either bound, fin is set to 1 to trigger a deviation test and prevent the model from further assessing the move.

The function returns the fin value of zero (not in violation) or one (in violation).

calcdev

The calcdev function calculates the total deviation of each period in a result to the average value for that result. It does this by finding the average (ave), taking the absolute value of the average minus the value for each period (individually, stored in variables dis1 through dis4), squaring each of those deviations and summing them.

The function returns that sum of squared deviations as dev.

updatecov, updateyield, updatecfs, and updateact

The update functions are called when a move is accepted to update key values of interest:

- The objective function value (updatecov)
- The results for gross revenue, net revenue, cost, sawlog material, and biochar feedstock (updateyield)
- The result for acres treated (updateact), and
- The result for composite resistance score (updatecfs).

Non-CRS values are updated by subtracting the values for the old treatment (unless the old treatment was the no-action alternative) and adding the values for the new treatment (unless the new treatment is the no-action alternative). CRS values are updated by adding the move effect variables se1 through se5 to the appropriate year in the cfs result. The values for all of these changes are taken directly from the accepted move effect. Each of these functions returns the updated result as a list, save for updatecov which returns the updated objective function value as a float.

savesol

When a move is accepted that generates a better objective function value than the best solution ever found, the savesol function is used to save the prescription for each fragment to a new solution. When constraints are enabled savesol will also be called to overwrite infeasible solutions with feasible solutions or solutions closer to feasibility regardless of the objective function value.

Regardless of the triggering event, savesol will always return the new solution as bsol.

loadsol

When Neo-Processor has run through its total move count, the best solution ever found (bsol) is loaded for further analysis using the loadsol function. The function moves through every fragment in the fragment dictionary (frags) and reassigns them to the prescription they had under the best solution ever found.

The function modifies each fragment in place and does not return an object.

calcdelivered

After Neo-Processor has run through its total move count and loaded the best solution ever found, the calcdelivered function is called to calculate the total biomass delivered in each period on a facility by facility basis. It does this by taking the sorted facility list (fcl) and creating a dictionary of results (deld) keyed by facility number. Once the dictionary and associated results are created, the function cycles through every fragment in the fragment dictionary (frags) and for each one

- Saves the stand identifier as st, fragment size as ac, and prescription as rx
- If any sawlog material is produced by that prescription, it looks up the sawlog facility and adds that biomass to the running total for that facility. And
- If any biochar feedstock is produced by that prescription, it looks up the biochar processing facility and adds that biomass to the running total for that facility.

After cycling through each fragment the function returns the completed delivered material dictionary (deld).

printout

The printout function is a general purpose function used for rapid diagnostic work. The function takes a list (olist) and file handle (fhandle) and prints each index in the list to the file in the file handle as a single tab delimited line followed by a new line character (\n). The function does not return an object or close the file handle.

REFERENCES

Bell, J.F. and Dilworth, J.R., 1988. Log scaling and timber cruising. OSU Book Stores. Inc., Corvallis, Oregon.

Denardo, Eric V. *Dynamic programming: models and applications*. Courier Corporation, 2012.

Dueck, Gunter. "New optimization heuristics: The great deluge algorithm and the record-to-record travel." *Journal of Computational physics* 104.1 (1993): 86-92.

Garber, Sean M., and Douglas A. Maguire. "Modeling stem taper of three central Oregon species using nonlinear mixed effects models and autoregressive error structures." *Forest Ecology and Management* 179.1 (2003): 507-522.

Hann, David W. "Revised volume and taper equations for six major conifer species in southwest Oregon." *Department of Forest Engineering, Resources, and Management. Oregon State University, Corvallis, Oregon. Forest Biometrics Research Paper 2* (2016).

Jain, T.B., J.S. Fried, R.F. Keefe, S. Loreno, C. Bell. 2017. Evaluating cost-effectiveness of multi-purpose fuel treatments in western mixed-conifer forests considering hazard, risk, longevity and co-benefits: Final Project Report to Joint Fire Sciences Program. [draft in preparation].