|  |
| --- |
|  |

**MUCP Algorithm MANUAL**

**by**

**Kirodh Boodhraj**

**Handed over to DFFE on 2 October 2025**

|  |  |
| --- | --- |
| Executive Summary | |
| Title: | MUCP Algorithm Manual (Kirodh Boodhraj) |
| Abstract: | MUCP Algorithms. |
| Keywords: |  |
| Objective: | Manual written for DFFE and future users. |
| Purpose: |  |
| Methodology: |  |
| Conclusions: |  |
| Limitations or Risks: |  |
| Benefits: |  |
| Contract Context: | This document is the result of a research effort funded by NRM and CSIR PG internal funding. |

|  |  |
| --- | --- |
| Approval Page | |
| Project Name | MUCP |
| Project No |  |
| Title |  |
| Prepared for |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Authored for | CSIR | Signature | Date |
| By | Kirodh Boodhraj | Kirodh | 30 Sept. 25 |
| Designation |  | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Change History | | | | |
| Change Control No | Date | Name | Description |
| Version 1 | 27 Jun. 23 |  | Original release. |
| Version 2 | 30 Sept. 25 |  | Cleaned up the document. |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Contents

[Abbreviations 4](#_Toc210040961)

[1 Introduction 5](#_Toc210040962)

[2 Get the code 6](#_Toc210040963)

[2.1 Install Git (if not already installed) 6](#_Toc210040964)

[2.2 Clone the MUCP Tool Repository 6](#_Toc210040965)

[2.3 Navigate into the Project Folder 7](#_Toc210040966)

[3 Downloading Python 8](#_Toc210040967)

[3.1 Check if Python is already installed 8](#_Toc210040968)

[3.2 Download Python 8](#_Toc210040969)

[3.3 Setting Up a Virtual Environment 9](#_Toc210040970)

[3.3.1 Create the virtual environment 9](#_Toc210040971)

[3.3.2 Activating the Virtual Environment 9](#_Toc210040972)

[3.3.3 Deactivating the Virtual Environment 10](#_Toc210040973)

[3.4 Quick Workflow Summary 10](#_Toc210040974)

[4 Setting up the environment (installing Python packages) 12](#_Toc210040975)

[5 Initial Program setup 13](#_Toc210040976)

[5.1 Preparing the Django Environment 13](#_Toc210040977)

[5.2 Applying Migrations 13](#_Toc210040978)

[5.2.1 Make migrations (create migration files if any new models were added or modified): 13](#_Toc210040979)

[5.2.2 Apply migrations (update the database structure): 13](#_Toc210040980)

[5.3 Creating a Superuser (Admin Account) 13](#_Toc210040981)

[5.4 Collecting Static Files (If Required) 14](#_Toc210040982)

[6 Running and accessing the viewer on your web browser 15](#_Toc210040983)

[7 Support and debugging 16](#_Toc210040984)

Abbreviations

|  |  |
| --- | --- |
| CSIR | Council for Scientific and Industrial Research |
| DFFE | Department of Forestry, Fisheries and Environment |
| MUCP | Management Unit Control Plan |
|  |  |
|  |  |
|  |  |
|  |  |

# Introduction

Le Maitre et al. (2012) developed a generic species and area prioritization model (MUCP Tool) for use in prioritising invasive alien plant control operations in South Africa using spatial data and Forsyth et al. (2012) . It schedules treatments of invasions in the catchment taking into account the current state of the invasions, benefits of clearing, treatments that are required and the resources provided in its budget. The MUCP tool does not generate a detailed annual schedule of annual operations but the DEA-Natural Resources Management (NRM) programme has an Annual Plan of Operations tool which serves that purpose.

This manual will guide the user on the algorithms constituting the MUCP Tool as well as the users input support data.

There are 5 main algorithms:

* Prioritization (arbritrary)
* Density (%)
* Person Days [normal and adjusted] (arbritrary)
* Costing (Currency)
* Flow (M^3/s)

Before the algorithms can be run, the users’ support data needs to be pre-processed.

The algorithms were coded in Python (version 3.11 latest at the time of this manual), using vectorized array calculations with Pandas, Numpy, Regex, and Geopandas in order to speed up the algorithm run times.

A full test case with approximately 400 rows for a 50 year plan runs in just under 10 seconds as a benchmark.

# User support data processing

# Prioritization

Perfect 👍 I’ll structure this as a \*\*technical manual section\*\* that you can drop directly into your MS Word document. It will include:

1. \*\*Theoretical Background\*\* (so your client understands the scientific/mathematical basis).

2. \*\*Detailed Implementation Steps\*\* (your step-by-step instructions reformatted into professional technical style).

3. \*\*Worked Example\*\* (using the Aggression and Elevation case you described).

4. \*\*Notes/Recommendations\*\* (e.g., about weights not needing to sum to 1).

---

# \*\*Technical Manual: Prioritization Algorithm\*\*

## 1. Theoretical Background

The \*\*prioritization algorithm\*\* is designed to combine multiple environmental and management variables into a single \*\*compartment prioritization score\*\*. This score determines the relative importance of a compartment for treatment, intervention, or monitoring.

The algorithm is based on \*\*multi-criteria decision analysis (MCDA)\*\* principles. Each variable (e.g., \*elevation\*, \*aggression\*, \*slope\*, etc.) is:

1. \*\*Mapped to a priority category\*\* (based on pre-defined ranges in the reference table).

2. \*\*Weighted according to user preferences\*\* (via the graphical user interface).

3. \*\*Aggregated across all selected variables\*\* to produce a final prioritization score.

### Key Properties

\* \*\*Weights do not need to sum to 1.\*\*

\* In practice, weights may sum to a value greater than 1 or less than 1.

\* However, for more stable results, the sum should preferably be \*\*greater than 1\*\*.

\* If the sum of weights is less than 1, a \*\*user warning\*\* should be generated to avoid under-representing priorities.

\* \*\*Prioritization Categories\*\* are discrete ranges that map raw data values (e.g., \*Elevation = 3\*) to integer priority levels (e.g., \*Priority = 1\*).

\* \*\*Final Prioritization Score\*\* is computed as the \*\*weighted sum of all category priorities\*\*.

$$

\text{Prioritization Score for Compartment } i = \sum\_{j=1}^{n} \big( P\_{i,j} \times W\_j \big)

$$

Where:

\* $P\_{i,j}$ = Priority category assigned to variable $j$ in compartment $i$.

\* $W\_j$ = User-defined weighting for variable $j$.

\* $n$ = Total number of prioritization variables.

---

## 2. Implementation Steps

The following procedure is applied to compute the prioritization score for each compartment:

1. \*\*Iterate Through Compartments\*\*

\* Read each compartment ID from the \*compartment prioritization CSV file\*.

\* Example: Compartment ID `H\_123`.

2. \*\*Loop Through Variable Columns\*\*

\* For each column (header) in the CSV file (e.g., `Elevation`, `Aggression`), process the variable individually.

3. \*\*Match Column Header to Reference Table\*\*

\* Cross-reference the column header against the \*\*Support Priority Fields Table\*\*.

\* Extract the internal representation of the variable name used by the tool.

\* Example: `"Elevation"` → `"elevation"`, `"Aggression"` → `"aggressive"`.

4. \*\*Retrieve User Weighting Values\*\*

\* Match the variable name to the \*\*user-selected weighting values\*\* in the GUI input file.

\* Example:

\* Elevation weighting = `0.9`

\* Aggression weighting = `0.2`

5. \*\*Assign Priority Category\*\*

\* For each variable, determine the \*\*priority category\*\* based on its raw value and the range definitions in the reference table.

\* Example category ranges:

| Elevation Range | Priority | Aggression Range | Priority |

| --------------- | -------- | ---------------- | -------- |

| 0–4 | 1 | 0–30 | 1 |

| 5–10 | 2 | 31–60 | 2 |

\* If Elevation = 3 → Priority = 1.

\* If Aggression = 50 → Priority = 2.

6. \*\*Multiply by Weighting\*\*

\* Multiply each category priority by the associated user weighting.

\* Example:

\* Elevation: $1 \times 0.9 = 0.9$

\* Aggression: $2 \times 0.2 = 0.4$

7. \*\*Aggregate Priorities\*\*

\* Sum across all weighted priorities for the compartment.

\* Example:

$$

\text{Final Score} = 0.9 + 0.4 = 1.3

$$

8. \*\*Assign Compartment Prioritization Score\*\*

\* Record the calculated score for compartment `H\_123`.

---

## 3. Worked Example

\*\*Input Data\*\*:

| Compartment | Elevation | Aggression |

| ----------- | --------- | ---------- |

| H\_123 | 3 | 50 |

\*\*Reference Table\*\*:

| Elevation Range | Priority | Aggression Range | Priority |

| --------------- | -------- | ---------------- | -------- |

| 0–4 | 1 | 0–30 | 1 |

| 5–10 | 2 | 31–60 | 2 |

\*\*User-Defined Weights\*\*:

| Variable | Weight |

| ---------- | ------ |

| Elevation | 0.9 |

| Aggression | 0.2 |

\*\*Step-by-Step Calculation\*\*:

1. Elevation = 3 → falls in range 0–4 → Priority = 1.

\* Weighted Score = $1 \times 0.9 = 0.9$.

2. Aggression = 50 → falls in range 31–60 → Priority = 2.

\* Weighted Score = $2 \times 0.2 = 0.4$.

3. Aggregate:

$$

\text{Final Score for H\\_123} = 0.9 + 0.4 = 1.3

$$

---

## 4. Notes and Recommendations

\* \*\*Weighting Guidance\*\*: While weights may sum to less than or greater than 1, a sum greater than 1 is recommended for balanced prioritization. Implement a \*\*user warning\*\* if weights sum to less than 1.

\* \*\*Flexibility\*\*: The algorithm supports any number of prioritization variables, provided corresponding category ranges are defined.

\* \*\*Reproducibility\*\*: Since the algorithm is deterministic, repeated runs with identical inputs will always yield the same prioritization scores.

\* \*\*Transparency\*\*: All input data (compartment values, reference tables, and weights) are stored in CSV/GUI input files, ensuring full traceability of results.

---

Would you like me to also \*\*extend this manual with a flow diagram\*\* (showing the decision steps from raw values → category mapping → weighting → final prioritization score)? That would make it more client-friendly for presentations.

# Density

Got it ✅

This is a complex algorithm, so I’ll expand it into a \*\*technical manual section\*\* similar to the prioritization one you asked for. The structure will be:

1. \*\*Theoretical Background\*\*

2. \*\*Variables Used\*\*

3. \*\*Detailed Algorithm Steps\*\*

4. \*\*Worked Example\*\*

5. \*\*Notes and Recommendations\*\*

This will be written in a way that you can \*\*copy into your MS Word manual\*\*.

---

# \*\*Technical Manual: Density Algorithm\*\*

## 1. Theoretical Background

The \*\*density algorithm\*\* calculates the effective stand density of invasive alien species within compartments, incorporating treatment history, species-specific reduction factors, and temporal propagation across budget years.

The algorithm is designed to:

\* Capture the \*\*interaction between MIU-linked species records and NBAL-linked species records\*\*, where applicable.

\* Apply \*\*initial and follow-up reduction factors\*\* depending on whether it is the first or subsequent treatment year.

\* Adjust densities based on \*\*species attributes (Idenscode, Age)\*\*, including mismatches between MIU and NBAL data.

\* Incorporate \*\*treatment frequency\*\*, allowing for multiple treatments per year or less frequent treatments across multi-year intervals.

\* Propagate densities across a \*\*20-year budgeting horizon\*\*, including densification effects where density increases between treatments.

The overall principle is that \*\*density in each year is derived from previous-year densities\*\*, reduced (or sometimes modified) by follow-up factors, with adjustments for mismatches between MIU and NBAL records.

---

## 2. Variables Used in Density Calculation

The algorithm draws on the following variables:

\* \*\*Treatment frequency\*\* (from compartment metadata).

\* \*\*Idenscode (species density index)\*\* from both:

\* NBAL linked species file.

\* MIU linked species file.

\* \*\*Age\*\* (e.g., Young, Adult) from both NBAL and MIU species files.

\* \*\*Initial reduction density factor\*\* (from species tab).

\* \*\*Follow-up reduction density factor\*\* (from species tab).

\* \*\*Budget year\*\* (from 1 to 20).

\* \*\*Densification factor\*\* (applied as a post-processing step each year).

Helper functions for propagation are defined as:

$$

\text{densify}(d, f) = d \times \frac{100 + f}{100}

$$

$$

\text{reduceDensity}(d, f) = d \times \frac{100 - f}{100}

$$

---

## 3. Algorithm Steps

### Step 1. Loop Through MIU Linked Species File

\* Group all entries by \*\*MIU compartment ID\*\*.

\* Example:

```

M\_H60B400169 Pinus pinaster 30 Adult

M\_H60B400169 Hakea sericea 20 Young

M\_H60B400169 Acacia longifolia 5 Young

```

### Step 2. For Each Compartment Entry

Check whether the MIU compartment ID matches an \*\*NBAL ID\*\* via the GIS mapping file.

#### Case A. \*\*NBAL Match Found\*\*

1. \*\*Idenscode and Age Match Exactly\*\*

\* Year 1: Use NBAL Idenscode.

\* Year ≥2: Use density from previous year.

\* Apply follow-up reduction factor:

$$

d\_{\text{new}} = d\_{\text{prev}} \times \frac{100 - F}{100}

$$

2. \*\*Idenscode Different\*\*

\* Apply \*\*halved follow-up fraction\*\*:

$$

d\_{\text{new}} = d\_{\text{miu}} \times \frac{100 - F}{100 \times 2}

$$

3. \*\*Age Different\*\*

\* Apply \*\*doubled follow-up fraction\*\*:

$$

d\_{\text{new}} = d\_{\text{prev}} \times \frac{100 - F}{100} \times 2

$$

4. \*\*Both Idenscode and Age Different\*\*

\* Add MIU and NBAL Idenscodes, then apply doubled follow-up fraction.

#### Case B. \*\*Only MIU (No NBAL)\*\*

1. \*\*Year 1\*\*

\* Apply initial reduction factor:

$$

d\_{\text{new}} = d\_{\text{idens}} \times \frac{100 - I}{100}

$$

2. \*\*Year ≥2\*\*

\* Use follow-up reduction factor with previous year’s density.

#### Case C. \*\*Compartment in GIS Only (No MIU/NBAL)\*\*

\* Assign density = 0.

---

### Step 3. Incorporate Treatment Frequency

\* Compute number of treatments per year:

$$

N\_t = \frac{12}{\text{Treatment Frequency (months)}}

$$

\* If $N\_t \geq 1$: repeat reduction algorithm \*\*$N\_t + 1$ times\*\* within a single year.

\* If $N\_t < 1$: invert treatment frequency (e.g., 0.5 → every 2 years). Apply reduction only once, then hold density constant until next scheduled treatment year.

---

### Step 4. Yearly Propagation

\* Continue propagation until \*\*Year 20\*\*.

\* Each year’s density is derived from the previous year’s density using follow-up factors and treatment frequency.

\* Apply densification factor at the end of each year:

$$

d\_{\text{year+1}} = \text{densify}(d\_{\text{year}}, f\_{\text{dens}})

$$

---

### Step 5. Special Rules

\* If MIU and NBAL \*\*age categories differ\*\*, double the density.

\* If NBAL present, multiply final result by a \*\*1.6 scaling factor\*\* (empirical correction).

\* If only MIU, no scaling applied.

---

## 4. Worked Example

### Input: MIU Linked Species

| Compartment | Species | Idenscode | Age |

| ------------ | ----------------- | --------- | ----- |

| M\_H60B400169 | Pinus pinaster | 30 | Adult |

| M\_H60B400169 | Hakea sericea | 20 | Young |

| M\_H60B400169 | Acacia longifolia | 5 | Young |

### Species Reduction Table

| Species | Initial Reduction (%) | Follow-up Reduction (%) |

| ----------------- | --------------------- | ----------------------- |

| Pinus pinaster | 50 | 20 |

| Hakea sericea | 50 | 20 |

| Acacia longifolia | 50 | 20 |

### Step 1. Initial Density (Year 1, MIU only)

\* Pinus pinaster: $30 \times (100 - 50)/100 = 15$

\* Hakea sericea: $20 \times (100 - 50)/100 = 10$

\* Acacia longifolia: $5 \times (100 - 50)/100 = 2.5$

\* \*\*Compartment Total\*\* = 27.5

### Step 2. Year 2 Density

\* Follow-up factor = 20%.

\* Each species density multiplied by 0.8.

\* Compartment Total = $27.5 \times 0.8 = 22.0$.

### Step 3. Year 3 Density

\* Compartment Total = $22.0 \times 0.8 = 17.6$.

### Step 4. Apply Densification Factor (e.g., 10%)

$$

d\_{\text{year+1}} = d\_{\text{year}} \times \frac{100+10}{100}

$$

\* Year 3 adjusted density = $17.6 \times 1.1 = 19.36$.

### Step 5. Continue to Year 20

\* Repeat yearly propagation using the same rules.

---

## 5. Notes and Recommendations

\* \*\*NBAL Adjustment Factor\*\*: Always apply 1.6 multiplier when NBAL data is available.

\* \*\*Mismatches\*\*: If Idenscode or Age mismatches occur, density fractions must be halved or doubled as specified.

\* \*\*Treatment Frequency\*\*: High-frequency treatments accelerate density reduction, while low-frequency treatments delay reductions across years.

\* \*\*Densification\*\*: Accounts for ecological rebound of invasive species between treatments. Must always be applied at the end of each year.

\* \*\*Long-Term Stability\*\*: Over 20 years, the density trajectory converges towards low values, provided follow-up factors remain significant.

---

# Person Days

Got it 👍 — I’ll draft this section for your manual in the same style as your \*\*prioritization algorithm documentation\*\*, but expanded into a \*\*formal, technical manual style\*\* so it’s clear, rigorous, and professional for your client.

I’ll split it into \*\*two sections\*\*:

1. \*\*Normal Person Days (Base Calculation)\*\*

2. \*\*Adjusted Person Days (Including Walking/Driving Time and Slope Adjustments)\*\*

and also provide \*\*theory/explanation + step-by-step algorithm + example\*\*.

---

# 📘 Person Days Algorithm

The Person Days (PD) algorithm provides a standardized framework for estimating the labor requirements (in person-days) associated with clearing operations in compartments and management units (MIUs). This algorithm replaces earlier approaches that lacked slope dependence and that did not properly account for travel time.

The new algorithm introduces two tiers of calculation:

\* \*\*Normal/Base Person Days\*\* – a direct calculation based on Person Days per hectare (PPD) and the area of the compartment.

\* \*\*Adjusted Person Days\*\* – an extended formulation that corrects the base values to account for travel overheads (walking and driving time) and slope effects.

This section describes both formulations, the underlying assumptions, and the computational workflow in detail.

---

## 1. Normal Person Days (Base Calculation)

### 1.1 Theory

Normal person days form the foundation of the calculation. They represent the idealized labor effort under standard working conditions, without explicit adjustment for terrain or travel time. The calculation is derived directly from the \*Clearing Norms Table\* where \*\*PPD (person days per hectare)\*\* values are prescribed for specific combinations of variables such as species growth form, treatment type, riparian/terrestrial classification, process stage (initial/follow-up), and density class.

The base person days for a compartment is expressed as:

$$

PD\_{\text{base}} = PPD \times A

$$

Where:

\* $PPD$ = Person days per hectare (from clearing norms)

\* $A$ = Area of the compartment (ha)

### 1.2 Workflow

1. \*\*Iterate over the GIS mapping file\*\*: For each compartment, check whether there is only a compartment ID, or whether MIUs (Management Inventory Units) and/or NBAL (Norm-Based Allocation Layer) records exist.

\* If \*compartment only\* (no MIU, no NBAL), no person days are assigned.

\* If MIU and/or NBAL are present, filter the associated MIU-linked species entries.

2. \*\*Apply Subroutine 1 (Treatment Method Selection)\*\*:

\* If only one treatment method exists, select it directly.

\* If multiple treatment methods exist, apply ordered selection rules:

\* Adult + Landscape → Ring bark

\* Adult + Riparian → Fell

\* If unavailable → Bark Strip

\* If still unavailable → Lopping/Pruning

3. \*\*Apply Subroutine 2 (Clearing Norm Filtering)\*\*:

Filter the clearing norms spreadsheet step by step:

\* Match density (Idenscode or previous year density).

\* Match size class (Age).

\* Determine riparian/land classification.

\* Match species → growth form → validate against support\_growth\_form table.

\* Filter process (Initial vs Follow-up).

\* Apply treatment method filter.

\* At this stage, one valid record remains.

4. \*\*Extract PPD\*\*: The record provides the Person Days per hectare.

5. \*\*Compute base PD\*\*: Multiply PPD × Area.

---

## 2. Adjusted Person Days

### 2.1 Theory

The adjusted calculation incorporates \*\*real-world operational factors\*\*:

\* \*\*Walking and Driving Time Overheads\*\*: Time lost due to travel reduces effective clearing time.

\* \*\*Slope Dependence\*\*: Steeper slopes require proportionally greater labor effort.

The adjustment is based on the following functional form:

$$

PD\_{\text{adjusted}} = PD\_{\text{base}} \times \frac{WD}{WD - 2 \times (T\_{walk} + T\_{drive}) / 60}

$$

Where:

\* $PD\_{\text{base}}$ = Base person days

\* $WD$ = Standard working day length (default = 8 hours)

\* $T\_{walk}, T\_{drive}$ = Walking and driving times (minutes)

\* The denominator represents effective working hours after subtracting travel overheads.

⚠️ Boundary conditions:

\* If $WD > 12$ or $WD < 4$, cap the values at 12 and 4 respectively.

Finally, slope adjustments are applied:

$$

PD\_{\text{final}} = PD\_{\text{adjusted}} \times f\_{\text{slope}}

$$

Where $f\_{\text{slope}}$ is a multiplier derived from the slope class of the compartment.

### 2.2 Workflow

1. \*\*Obtain Base Person Days\*\* as described in Section 1.

2. \*\*Apply Adjustment for Travel Time\*\*:

\* Compute effective hours using walking + driving times.

\* Modify PD according to the formula above.

3. \*\*Apply Slope Adjustment\*\*:

\* Look up slope range for the compartment.

\* Multiply adjusted PD by the slope factor.

4. \*\*Aggregate Across MIU-linked Species\*\*:

\* For each MIU species entry, compute adjusted PD.

\* Sum all adjusted PD values to get total per compartment.

---

## 3. Example Calculation

### Example Inputs

\* Compartment ID: \*\*H\_123\*\*

\* Area = 15 ha

\* PPD = 0.35 person days/ha (from clearing norms)

\* Working day (WD) = 8 hours

\* Walking time = 20 min

\* Driving time = 30 min

\* Slope factor = 1.3

### Step 1: Base Person Days

$$

PD\_{\text{base}} = 0.35 \times 15 = 5.25

$$

### Step 2: Travel Adjustment

$$

PD\_{\text{adjusted}} = 5.25 \times \frac{8}{8 - 2 \times (20+30)/60}

$$

$$

= 5.25 \times \frac{8}{8 - 2 \times (50/60)}

$$

$$

= 5.25 \times \frac{8}{8 - 1.67} = 5.25 \times \frac{8}{6.33} \approx 6.63

$$

### Step 3: Slope Adjustment

$$

PD\_{\text{final}} = 6.63 \times 1.3 \approx 8.62

$$

### Result

\* \*\*Normal Person Days\*\* = 5.25

\* \*\*Adjusted Person Days\*\* = 8.62

---

# Costing

# Flow

# Main costing loop

# Release notes & references

## Release notes

The open-source implementation of the MUCP tool introduces several important differences compared to the previous (proprietary) version. These differences affect both the underlying algorithms and the functionality available to users.

### Algorithmic Differences

#### Prioritization and Density

The implementations of the prioritization and density remain consistent with the previous tool. This ensures continuity of results where these parameters are concerned, so outputs based on these algorithms are directly comparable between the two systems.

#### Person-Days Calculation

The person-days algorithm has undergone a significant change.

In the previous tool, the calculation did not account for slope and deviated from the method provided in the official Person Days Excel spreadsheet provided by DFFE.

In the open-source tool, the algorithm has been corrected and now explicitly incorporates slope dependence. This results in more realistic estimates of the effort required, especially in areas with non-trivial terrain gradients.

#### Costing Algorithm

The costing (budgeting) algorithm continues to use prioritization, density, and person-days as the core inputs. However:

Because the person-days calculation has been corrected, the overall costing outputs will differ from the previous tool.

Depending on compartment-specific parameters such as slope, area, and effort requirements, costs may be higher or lower than those reported by the previous version.

This adjustment results in outputs that are better aligned with real-world field conditions and financial planning.

#### Flow algorithm

The flow algorithm underwent a slight change from the previous tool which did not have a riparian factor adjustment present. The riparian depending if it is landscape or riparian, needed a factor of 1.5 adjustment. This is now incorporated into the new algorithm set.

This means that the flow results from the previous tool will be different than the current opensource tool. Note also that the flow would have changed due to the person day algorithm being updated with the slope factor correction.

### Algorithm Derivation Methodology

The algorithms used in the opensource version were mainly reverse-engineered through brute-force testing of the desktop tool. This involved painstaking sensitivity analyses, where individual input values were varied systematically, and the resulting outputs were analysed to infer the governing equations.

As a result, the algorithmic logic was reconstructed based on observed tool behaviour rather than pre-existing documentation.

While the derived algorithms reproduce the functional behaviour of the original tool with a high degree of fidelity, there may be minor differences due to the lack of direct access to the original formulations.

### Functional and Structural Differences

#### User-Defined Prioritization Categories

The previous tool limited users to a fixed set of prioritization categories. The open-source version removes this limitation, allowing users to define their own prioritization variables and categories. This provides greater flexibility for tailoring the tool to local or project-specific management needs.

#### Extended Budget Horizons

Budgets are no longer restricted to a 20-year planning horizon. Users can project costs and treatment plans for arbitrary timeframes, supporting both long-term strategic planning and shorter adaptive management cycles.

#### Herbicides and Cost Recovery

As in the previous version, herbicide usage and cost recovery mechanisms have not yet been implemented. These remain placeholders for potential future enhancements.

#### Corrections to Column Names, Units, and Terminology

Several column names, units, and descriptive terms from the previous tool were found to be inconsistent or unclear. These have been corrected and standardized in the open-source implementation, ensuring greater clarity and consistency of outputs.

### Technical Implementation

#### Open-Source Codebase

The tool has been re-implemented in Python and is fully open-source.

The algorithmic core is implemented primarily using the Pandas package for high-performance data manipulation and calculation.

The user interface and visualization components are delivered through a Django-based web application, supported by auxiliary Python packages.

This separation of concerns between algorithms and the web viewer ensures modularity and makes it easier to extend, maintain, and audit the system.

### Beta Release

This release should be considered beta software. While the tool is functional and provides scientifically valid outputs, there may still be bugs or edge cases that need to be resolved. User feedback during this stage is critical to refine and stabilize the system.

## References

Some references that were useful in figuring out the MUCP algorithms.

* <https://scholar.sun.ac.za/bitstream/handle/10019.1/123841/boast_cost_2021.pdf?sequence=2&isAllowed=y>
* <https://sites.google.com/site/wfwplanning/planning>
* Greg Forsyth’s manual found in the link above
* Andrew Wannenburgh’s Person Days Excel spreadsheet
* David le Maitre’s email with the Flow algorithm and riparian 1.5 factor adjustment correction

Manual End