RS_CONTAIN

Customised version of RangeShifter v2.0 for

Adaptive Management of Invasive Non-native Species

User Manual

Stephen C. F. Palmer

University of Aberdeen

December 2022

1 Introduction

This manual describes the new features provided in RS_CONTAIN, a customised version of RangeShifter v2.0 (RS; Bocedi et al. 2021) for modelling adaptive management (AM) of invasive non-native species (INNS). It should be used in conjunction with the current version of the RS User Manual, which may be obtained from https://rangeshifter.github.io/.

Users are strongly advised to familiarise themselves with RS and to undertake its tutorials before using RS_CONTAIN.

RS_CONTAIN is provided as two executable files, each of which is compiled for Windows machines with a 64-bit processor: the batch version (*RSbatch_v2_0_CONTAIN_64bit.exe*) and the graphical user interface (GUI) version (*RS_v2_0_ CONTAIN_64bit.exe*).

The new features comprise the AM module itself, dispersal by animal vectors, dispersal by wind and habitat-dependent demography. Each feature is optional, and they may be applied in combination (except that the two new dispersal models are mutually exclusive) and with any appropriate existing options in RS.

2 Adaptive Management

2.1 The Damage Landscape

An additional landscape layer may be read in which the locations of potential damage sites are specified. This damage layer must have the same cell size, origin and extent as the habitat layer. Any cell containing a positive integer indicates a location of potential damage (financial, social, environmental, etc.). The cell value defines the maximum level of damage possible at that location, and maximum damage values define the relative importance of damage cells to each other in arbitrary units. Otherwise, cells containing zero indicate no damage can occur there.

In the GUI, the name of the text file holding the damage map is supplied through the *Landscape* form following selection of the habitat raster and, optionally for a patch-based model, the patch raster. The option *Load damage map* is checked by default, and must be unchecked in order to run the model without a damage layer. In the batch version, the name of the damage map file is supplied in the *LandFile* as an additional *DamageFile* column following the *SpDistFile* column. To run the model without a damage layer, set *DamageFile* to NULL.

Additionally, the *Landscape* form of the GUI (by checking the *Load exclusion map* box) and the batch *LandFile* (by entering the map file name in the *ExclusionFile* column) permit the optional specification of a cull exclusion map (see next section), which also must have the same cell size, origin and extent as the habitat layer.

2.2 Management Control and Damage Caused

2.2.1 Management Cull

Details of where, when and how the INNS is to be controlled in the model, assumed to take the form of a cull (i.e. complete removal of all or a proportion of the INNS present in one or more cells/patches in the current year), are specified in the *Management Control* form of the GUI (Figure 2.1) or the *ManagementFile* of the batch version (the name of which is given in the *Control* file following the *GeneticsFile* line; if *ManagementFile* is NULL then no cull is applied).

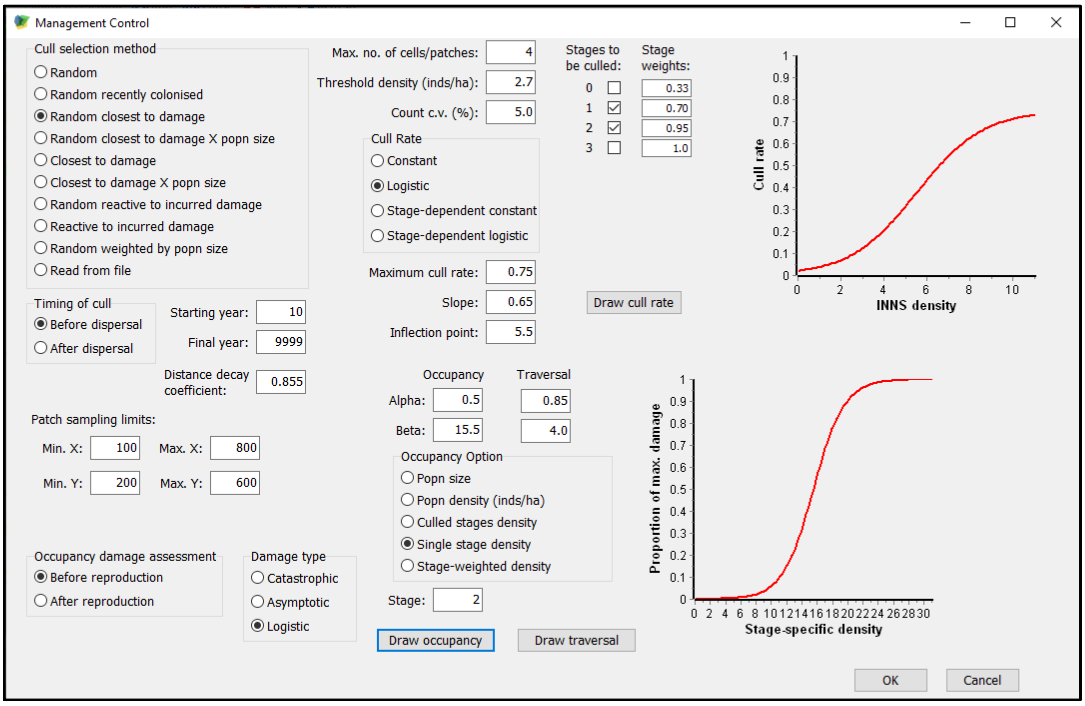

Figure 2.1. The *Management Control* form.

The method by which local populations of the INNS are selected for culling from all the qualifying local populations (see below) may take one of ten options:

at *random*

at *random* but biased towards those most *recently colonised*

at *random* but biased towards local populations in patches *closest to damage* locations

at *random* but biased towards local populations in patches *closest to damage* locations and in relation to the current local *population size*

deterministic and biased towards local populations in patches *closest to damage* locations

deterministic and biased towards local populations in patches *closest to damage* locations and in relation to the current local *population size*

at *random* but biased towards locations of recent *incurred damage*

deterministic and biased towards locations of recent *incurred damage*

at *random* but biased in relation to the current local *population size*

read from an input text file

The options relating to damage locations are permissible only if a damage map has been loaded, in which case each cell/patch is assigned a damage proximity index I_D derived from Hanski's index for connectivity in a metapopulation:

$$I_D = \sum_{i=1}^{n} D_i e^{-\alpha d_i}$$

where D_i is the maximum damage level for damage location i, d_i is the Euclidean distance between the centroid of the patch (or cell) and damage location i and α is a specified constant *distance decay coefficient*. If damage location i lies within the patch then d_i is taken to be zero regardless of its relationship to the centroid so that the damage location makes the maximum possible contribution to the damage index of the patch.

The random biased options take the form of a weighted lottery in which the probability of success (i.e. being selected to be culled) is directly proportional to the bias criterion. For the deterministic options, the cells/patches are ranked according to the bias criterion, and are selected for culling starting from the highest ranked and continuing down the ranking until the required number has been selected.

If the option *Read from file* is selected, click on the *Read cull selection file* button to select a file which lists specific combinations of year and cell/patch which are to be selected for culling (see the documentation in *ManagementFile.xlsx* for the format of the *CullFile*). Note that all other selection options (see below) except the maximum number of cells/patches remain in force; for example, if a specified patch in the specified year falls short of the threshold density, then no cull will occur there in that year.

The *Timing* of the cull may be either *before dispersal* (but after reproduction) or immediately *after dispersal*. A *Starting year* may be set, before which no cull will occur, thereby allowing the population to reach equilibrium after initialisation before culling commences. Similarly, a *Final year* may be set, after which no cull will occur, thereby allowing the population potentially to recover. Unless selection is read from a file, a *Max. no. of cells/patches* to be culled must be specified, and if the number of qualifying patches falls short of this maximum in the current year, then all qualifying patches are culled. Optionally (except if *Read from file* is selected), the cull may be restricted to a rectangle defined by the *Cell sampling limits*, which are expressed in terms of row and column numbers (not real-world co-ordinates). The equivalent restriction *Patch sampling limits* for a patch-based model permits selection of any patch which at least partially falls within the rectangular area. Further, the cull may be restricted (except if *Read from file* is selected) more specifically by providing a cull exclusion map. Any cell containing a positive integer in this map is excluded from the management cull; cells not to be excluded must be set to zero. In a patch-based model, a patch is excluded from the cull if any one of its cells is excluded. Note that exclusion takes precedence over all other selection methods, including the limits rectangle.

An occupied patch qualifies to be potentially included in the current year's cull if its local population exceeds a *threshold density* (inds/ha), which must be set to be greater than zero, but may be set to a very low value if all occupied patches are to be regarded as qualifying for the cull. However, it is recognised that in reality the local population density may not be known accurately, and therefore an option is provided to estimate the local population density from a count sampled at random from a normal distribution centred on the true count and having a standard deviation calculated from a *Count c.v.* (coefficient of variation) common to all patches. Note that in a stage-structured model, the local population density is based on the specified *Stages to be culled* (of which there must be at least one selected), which may be less than the density attained by the whole local population. Also, for a stage-structured population, a weight (between 0.0 and 1.0 inclusive) must be set for each stage to allow stages (whether selected to be culled or not) to be given weightings other than the default of zero when the incurred damage differs between stages (see below). Thus, for example, a sub-adult might be given half the weighting of an adult if it is considered that sub-adults cause less damage than adults.

The actual size of the cull within the selected patches in the current year is determined by the *Cull rate* (the proportion of individuals which will be removed). This is the expected rate of removal of those individuals of the selected stage(s) present in the selected local population(s) at the time of the cull, and it may be *Constant* or a *Logistic* function of the INNS density in the patch. In the latter case, the cull rate p_c is given by:

$$p_c = \frac{p_{max}}{(1 + e^{-(d-\beta)\alpha})}$$

where p_{max} is the maximum cull rate, d is the local density of those stages which are to be culled, β is the inflection point and α is the slope. The cull rate is implemented as a probability of removal applied independently to each individual; thus, where the local population size is small, the achieved cull rate in the current year may differ substantially from the expected rate.

Further, if the INNS population is stage-structured, and two or more stages have been selected for culling, the cull rates may be *Stage-dependent constant* or follow distinct *Stage-dependent logistic* functions of density. In that case, the *Stage-dependent cull rates* option will be presented, which, when clicked, will open a new window (Figure 2.2) in which the parameters of separate logistic functions (as above) for each culled stage may be entered. For the *Stage-dependent constant* option, only the *Rate* is required for each stage; the second and third columns of the table are not applicable, and are headed *N/A*.

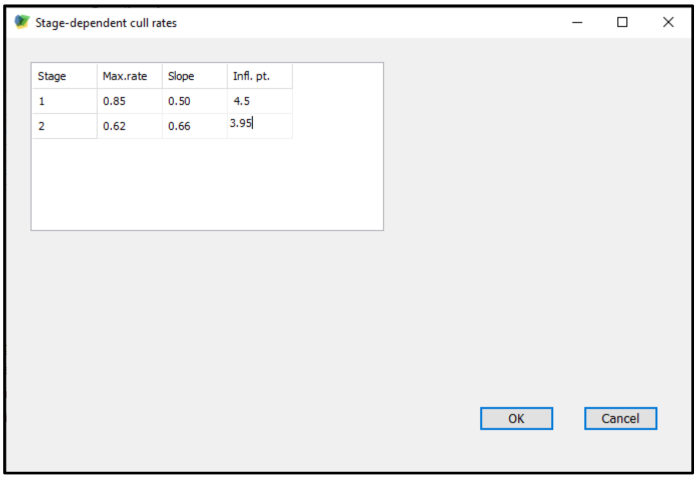

Figure 2.2. The *Stage-dependent cull rates* form.

2.2.2 Damage caused by the INNS

When the model is run, damage can occur in one of two ways. If a damage cell is located within a suitable patch (or cell) for an INNS population, then 'occupancy damage' occurs in a particular year if the cell is occupied by the INNS. Any damage cell, including those which lie within the matrix (i.e. not coincident with a suitable patch for the INNS), may incur 'traversal damage' only if the transfer model is the stochastic movement simulator (SMS), in which case the level of traversal is the total number of times dispersers have moved through the cell in the current year.

The actual level of damage incurred in the current year within a damage cell ranges from zero up to the maximum damage for the cell, and is a function of the local INNS population for occupancy damage and/or the total number of traversals for traversal damage (if both occur, they are treated additively). The assessment of occupancy damage may occur either *Before reproduction* (when there are no juveniles in the population) or *After reproduction* (in which case juveniles may contribute to the level of damage). The relationship may take one of three functional forms:

Catastrophic - in the event of any non-zero local population or level of traversal, the maximum damage occurs (i.e. a binomial state of no damage or maximum damage);

Asymptotic - the level of damage D is given by the Monod equation

$$D = \frac{D_{max}X}{(X + \beta)}$$

where D_{max} is the maximum possible damage, X is the local population or number of traversals and β is the half saturation constant (the level of X at which D is half of D_{max});

Logistic - a sigmoidal relationship (similar to the Holling type III functional response) in which the level of damage D is given by

$$D = \frac{D_{max}}{(1 + e^{-(X-\beta)\alpha})}$$

where D_{max} is the maximum possible damage, X is the local population or number of traversals, β is the inflection point and α is the slope.

For the purpose of calculating occupancy damage, the local population may be treated as (0) the total population size, (1) the total population density (to allow patches of different size in a patch-based model to be treated similarly), (2) the (unweighted) density of individuals in the stages to be culled, (3) the density of individuals in a specified stage or (4) the stage-weighted densities of all individuals. Options (2), (3) and (4) apply to a stage-structured population only.

In any one model, the same functional form is applied to both occupancy and traversal damage. The GUI provides a facility for drawing on the screen the form of the relationship between D and X given the value of the function parameter(s), and it is recommended that it be checked visually prior to running any new model in batch mode.

3 Dispersal Models

3.1 The 2Dt kernel for dispersal by animal vectors

The 2Dt kernel (Clark et al. 1999) models the statistical distribution of seeds with distance from the source (usually the parent tree), such that close to the source the distribution is convex but at larger distances it is concave, a pattern which has been found to fit observed data for seeds dispersed by animals better than other statistical kernels such as the negative exponential.

In the GUI version, select the *2Dt* option for *Dispersal Kernel* on the *Dispersal* tab of the *Species Parameters* form (Figure 3.1). This will enable the parameters for two 2Dt kernels to be specified, along with the proportion of seeds which is distributed according to the first kernel (set this to 1.0 if only one kernel is to be applied). For each kernel, a scaling parameter (*U0*) and a shape parameter (*P0*) must be specified. Note that these are the estimates that would be obtained from fitting a generalised linear model to log-transformed seed-trap data (Powell and Aráoz, 2018), and they are transformed to the parameters *u* and *p* as described by Clark et al. (1999) as:

$$u = e^{U0} \text{ and } \quad p = e^{P0}$$

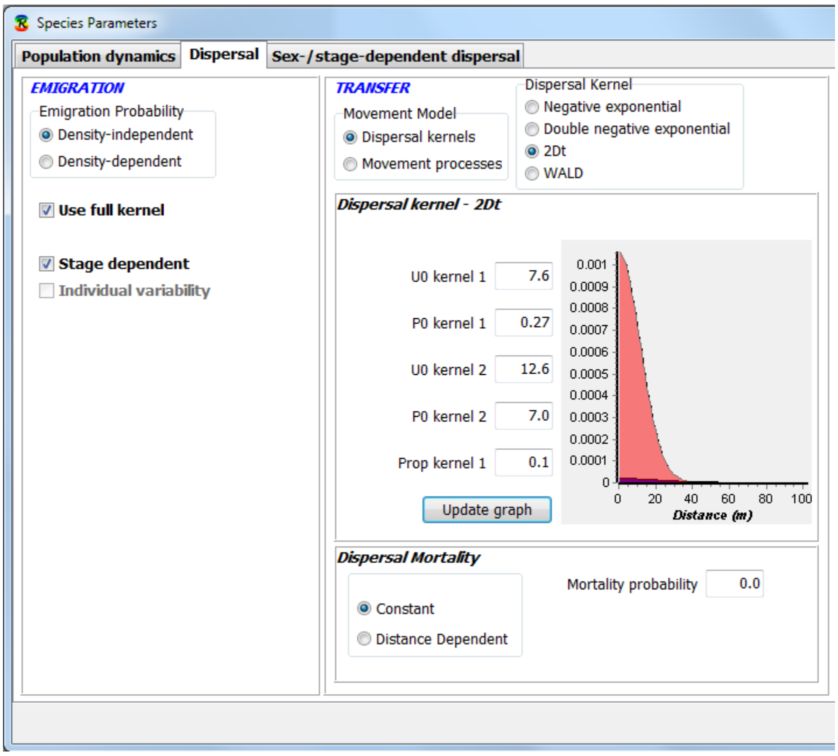

Figure 3.1. Specifying the 2Dt dispersal kernel.

In the batch version, set the value of the *Transfer* parameter to 3 in the *CONTROL.txt* file to specify the 2Dt kernel, and set the kernel parameters in the *TransferFile* according to the format described in the 2Dt kernel sheets of the *TransferFile.xlsx* documentation file.

3.2 *The WALD kernel for dispersal by wind*

The WALD model is an analytical model for predicting the dispersal kernel of wind-dispersed seeds (Travis et al. 2011, Caplat et al. 2012). The shape of the kernel depends on attributes of the parent tree (canopy height and seed release height), attributes of the seed (terminal velocity) and on prevailing wind conditions (horizontal and vertical velocities and turbulence).

In the GUI version, select the *WALD* option for *Dispersal Kernel* on the *Dispersal* tab of the *Species Parameters* form (Figure 3.2). This will display a panel, on which should be specified the mean horizontal wind speed (*Mean U*, m/s), the standard deviation of the vertical wind speed (*Sigma_w*, m/s), the canopy height (*Hc*, m), the seed terminal velocity (*Vt*, m) and the dimensionless turbulence coefficient (*Kappa*), all of which are required to be greater than zero. For each non-juvenile stage, whether or not its fecundity is non-zero, a seed release height (*Hr*, m) is required, which must be greater than zero and no more than the canopy height. It is also necessary to specify the mean wind direction (towards which the wind mostly blows in degrees, from 0° for due north clockwise to 359.999° for just west of north) and its standard deviation (degrees).

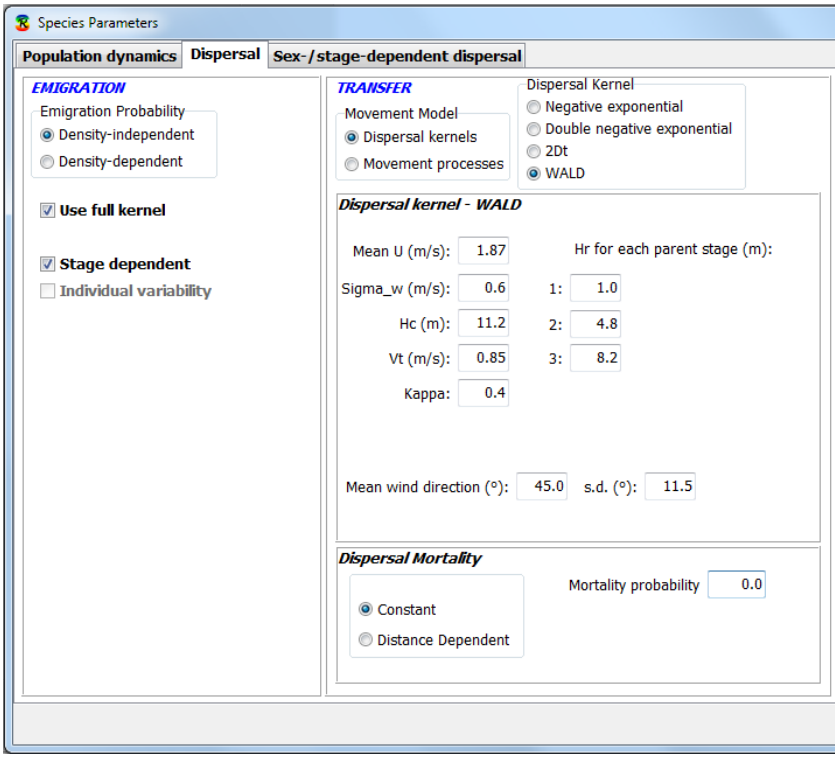

Figure 3.2. Specifying the WALD dispersal kernel.

In the batch version, set the value of the *Transfer* parameter to 4 in the *CONTROL.txt* file to specify the WALD kernel, and set the kernel parameters in the *TransferFile* according to the format described in the WALD kernel sheets of the *TransferFile.xlsx* documentation file.

Note that this option implicitly assumes that a seed dispersed by wind can land anywhere, including in a 'no-data' region (such as a lake) or beyond the landscape boundary, since dispersal by wind is non-selective in terms of the settlement location. The landscape boundary is therefore implicitly 'absorbing', regardless of the option applied in the model. Seeds landing in a 'no-data' cell immediately die, and are given a status code of 19 in the output *Inds* file.

4 Habitat-dependent Demography

This option enables vital demographic rates (e.g. the establishment and survival of tree seedlings) to differ between habitat classes, and is particularly applicable for a dynamic landscape in which there is a limited time period during which establishment may occur, for example following some form of disturbance or land-use change. The option has been implemented only for species having a stage-structured life-cycle, i.e. those for which a Leslie transition matrix is specified to set demographic rates (fecundity, development and survival), and may be chosen only if the landscape raster comprises discrete habitat codes (*LandType* 0 in the batch version).

In the GUI version, check the *Habitat-dependent transition matrices* option on the *Population dynamics* tab of the *Species Parameters* form, and the click on the *Set transition matrices* button (Figure 4.1).

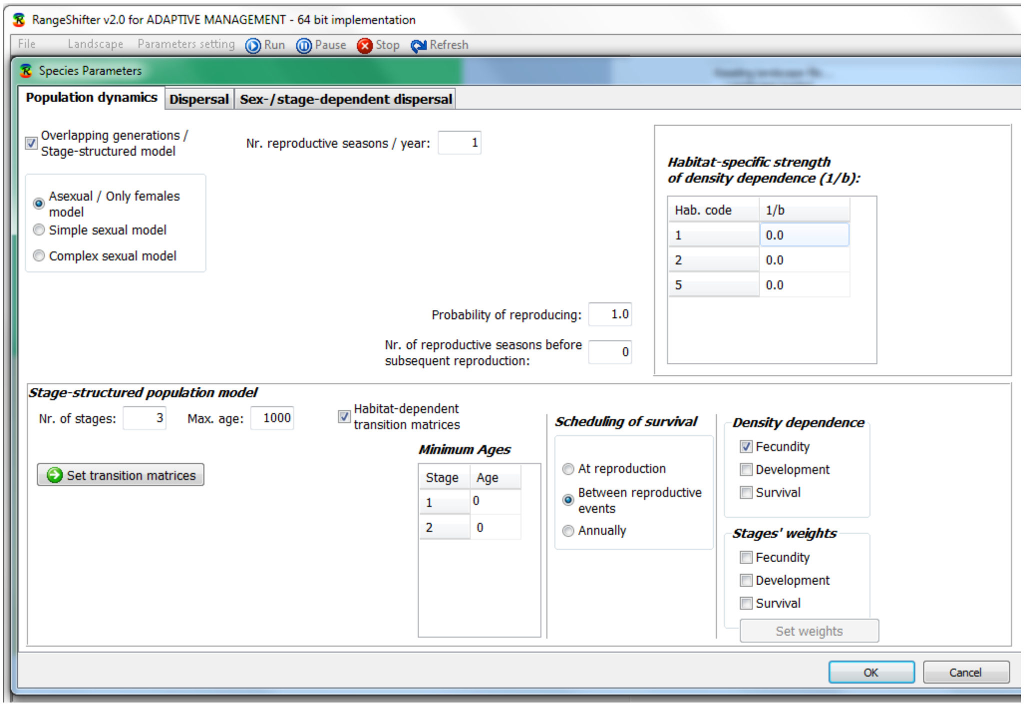

Figure 4.1. Selecting the option for habitat-dependent demography.

The *Transition Matrices* form (Figure 4.2) will open, on which up to ten habitat codes will be listed. Note that transition matrices may be applied to the first ten habitats only; it should therefore be ensured that suitable habitat classes have one of the ten lowest codes, reserving higher codes for unsuitable land classes occurring within the inter-patch matrix. Initially, the file name against each habitat code will be *NULL*. Enter a valid *Habitat code* and click on the *Select File* button to choose a transition matrix file for that habitat. The transition matrix file takes the standard format required for the batch version of RS (see *StageStrucFile.xlsx*). If you make a mistake, the *Remove File* button may be used to reset a file name to *NULL*. Once all the required transition matrices have been selected, click on the *OK* button to close the form and load the transition matrix data.

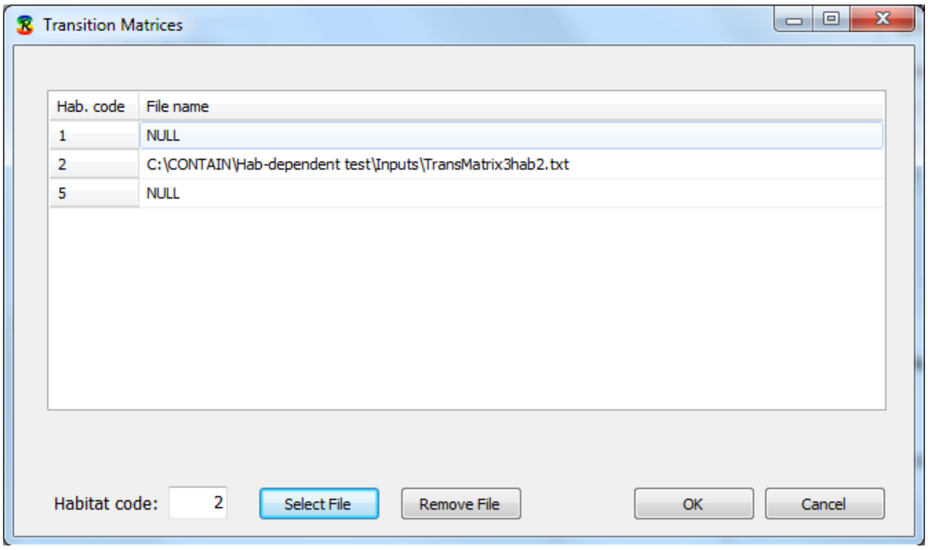

Figure 4.2. Selecting transition matrix files for habitat-dependent demography.

In the batch version, there is an additional *HabDemFile* column in the *StageStructFile*, which must either be *NULL* (in which case habitat-dependent demography is <u>not</u> applied and a *TransMatrixFile* must be specified) or hold the name of a file which contains a list for each suitable habitat code of the corresponding *TransMatrixFile* (see *StageStrucFile.xlsx*). Note that in the batch version, as habitat codes must be sequential integers starting from 1, suitable habitats must be coded from 1 to 10.

An important point to remember in both GUI and batch versions is that a habitat code for which a transition matrix has not been specified has all demographic rates set to zero. **Therefore you should ensure that transition matrices and non-zero *1/b* values (*Kn* in the batch *ParameterFile*) correspond exactly** - otherwise, local populations in habitats having no transition matrix will go extinct immediately.

5 Output

If the output *Pop* file is selected, then an output *Cull* file is also produced with the same starting year and frequency as the *Pop* file. The *Cull* file lists the total number of individuals removed from each patch (or cell) at the specified time, and, if the population is stage-structured, also the number culled from each stage. If greater detail in terms of the age or sex of the culled individuals is required, then the *Inds* file must be produced, in which culled individuals are identified by a new *Status* code of 10.

If a damage map has been loaded, then an output *SummDamage* file is produced. It provides a summary index of damage incurred (occupancy and traversal combined) over the whole landscape on an annual basis. If a more detailed record of location-specific damage is required, then the *Damage Indices* option on the *Simulation Parameters* form may be selected (or in batch mode, by setting the *OutStartDamage* and *OutIntDamage* columns of the *ParameterFile*). The resulting *Damage* file lists the damage incurred at the specified time for each damage location in the landscape.

6 Tutorials

6.1 *Generic introduction*

In this example we will use the data from the second tutorial of RS to illustrate how to read a damage layer into RS_CONTAIN and explore how various management control options may act to reduce the incurred damage over time and space.

6.1.1 *Estimating total damage in the absence of management control*

Open the GUI version of the program, *RS_v2_0_CONTAIN_64bit.exe*, and set the working directory to the *Tutorial1* folder.

Open the *Landscape / Import Raster* window. Recall that the landscape for this exercise contains discrete habitat codes at a resolution of 10 m, and is patch-based. Note that there is a new check-box *Load damage map*, which is checked by default, and should remain so for this exercise (but leave the check-box *Load exclusion map* unchecked). When you click on *Import Landscape*, you will be prompted to select three landscape raster files from the *Inputs* folder: the habitats (*landscape_10m.txt*) and patch (*woodland_1ha_patchIDs.txt*) raster files as in RS, and lastly the damage layer (*damage_tutorial1_v01.txt*). Once these files have been selected correctly, the landscape will be displayed on the screen. Scroll the landscape panel slightly to the right, and on it you should see eight black squares, each within a separate (green) woodland patch, and three black rectangles (Figure 6.1). These indicate respectively the eight locations of potential damage within suitable INNS patches and three locations within the inter-patch matrix. The sizes of the black areas show where the damage cells are located, but not the maximum damage level in each cell.

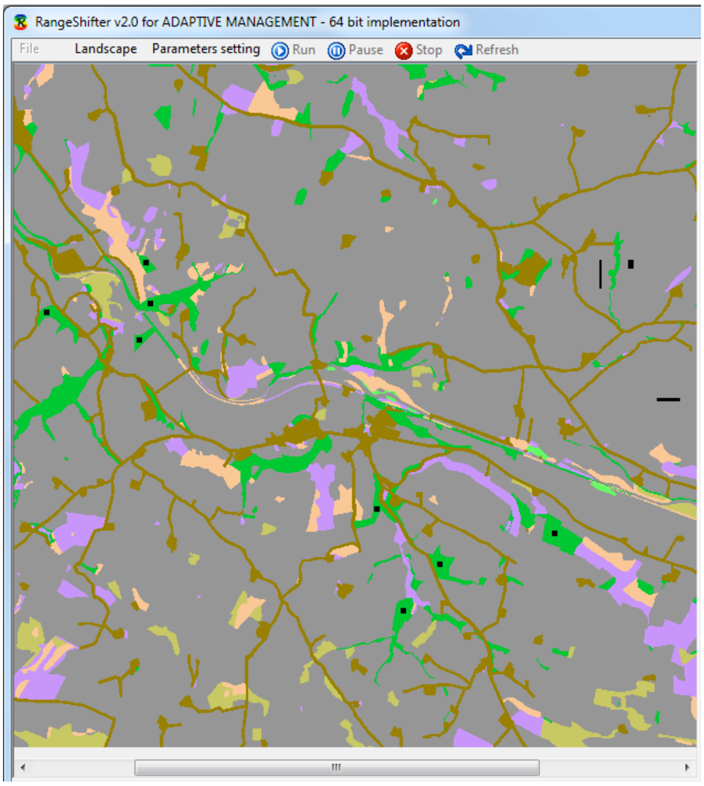

Figure 6.1. The locations of potential damage (black cells).

Now set up the *Species* parameters, which are similar to those for RS tutorial 2, except that *1/b* should be 5 inds/ha, the SMS *Step mortality* should be set to 0.005, the SMS *Memory size* set to 2 and settlement should be *Sex dependent* in that males are required to find a mate but females are not. The settings of all species parameters are listed in the file *Tutorial1_Parameters.txt*.

Next, open the *Management Control* window, in which are specified parameters for the INNS culling regime and for recording the damage incurred. In this first part of the tutorial, we will run the model with no control of the INNS, but certain parameters on this window must still be specified. Under *Stages to be culled*, check the box for stage 2, and set *Stage weights* to 1.0 for all three stages. In order that no culling occurs, change the *Threshold density* to 1000.0 inds/ha and the *Cull rate* to 0.0. Select the *Asymptotic* option for *Damage type*, and then set the *Occupancy Option* to *Popn density*. The default values for the *beta* parameters may be retained, but click on the *Draw occupancy* button to see how the level of occupancy damage (within an INNS suitable patch) is related to the local population density, and click on the *Draw traversal* button to see how the level of traversal damage (within the matrix) is related to the number of times dispersing individuals pass through a matrix cell.

Finally, set the *Simulation Parameters* for the model. You should run five replicates each of 30 years, and produce *Range*, *Populations* and *Damage Indices* outputs every year. All suitable patches should be initialised at 6 individuals/ha, of which 30% should be stage 1 and 70% should be stage 2.

Once the simulation has completed, the *Total damage incurred* graph will be drawn. This shows the combined total of all occupancy and traversal damage each year, averaged over the five replicates, together with a 95% confidence interval (dotted lines). The mean total damage should vary between about 700 and 850 units during the 30 year period.

Now check the output *Cull* file, which has been produced automatically together with the *Pop* file. The *Nculled* column should be zero throughout all replicates, i.e. no culling occurred. If that is not the case, check the *Management Control* options, and re-run the simulations.

Next, take a look at the *SummDamage* output file, and use whatever software you prefer (e.g. R, Excel) to calculate the mean damage each year across replicates, which should match the figures depicted in the *Total damage incurred* graph. Also calculate the total damage across all years for each replicate and then the mean across replicates: the mean should be somewhere in the region of 23000 units, and that is the figure we look to reduce substantially by applying management control to the INNS population.

6.1.2 Introducing management control

Refresh the model and re-open the *Management Control* window. We will firstly look at selecting patches to be culled at random (the default, which should already be selected), and assume that our resources allow up to 5 patches to be culled each year, but only if the INNS *Threshold density* is 1.0 inds/ha or more. Our ability to estimate the density is not perfect, and therefore we set the *Count c.v.* to 20%. Moreover, the culling efficiency is low at low density and much higher at high density; we allow for this by selecting the *Logistic* option for *Cull Rate*, and setting the *Maximum cull rate* to 0.85, the *Slope* to 2.0 and the *Inflection point* to 2.5 inds/ha. Use the *Draw cull rate* button to show the relationship between cull rate and INNS density on the screen (Figure 6.2).

Change the *Simulation number* to ensure that the results of this scenario do not overwrite those of the previous status quo scenario, and run the model.

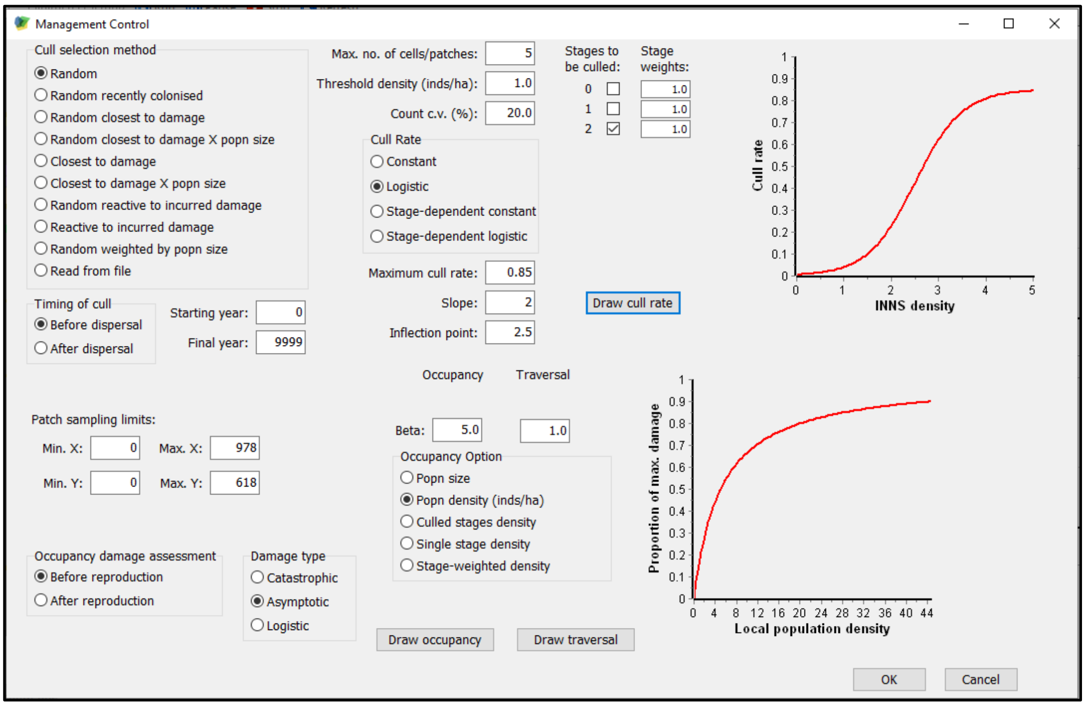

Figure 6.2. Displaying the relationship between cull rate and INNS density.

Again, firstly check the output *Cull* file, and select the records which have 1 in the *Selected* column. There should be five records for each year, which record the patches selected and how many individuals were culled in each. Sum the total number culled over 30 years within each replicate: it will vary considerably, but should be around 1400 to 1650 individuals over the 30 years. Much of that variation is due to variation in the number culled within selected patches each year. Inspection of the selected patches will show that the number culled varies substantially from none up to maybe 60 or more. Clearly, selecting patches and then failing to cull any individuals is an inefficient deployment of limited resources, especially as the culling rate is very low at low population density.

However, random selection of patches does have some effect, as the *SummDamage* output file shows. The total damage incurred should be reduced to around 20000 to 22000 units in each replicate, on average something like a 10% reduction compared with the no culling scenario.

6.1.3 Finding the best management control method

Refresh the model, re-open the *Management Control* window, and change the *Method* to *Random closest to damage*. This option selects patches for culling on the basis of a random lottery weighted towards patches close to damage locations, and requires an additional parameter, the *Distance decay coefficient*, which determines the strength of the bias towards patches close to damage locations (a large coefficient results in high bias); set the coefficient to 0.05, which results in a half-distance of about 15 cells.

Remember to change the *Simulation number* to a new value before running the model.

Firstly, examine in the output *Cull* file which patches have been selected for culling each year. You should find that patches 23 to 27, 37, 40, 41 and 44 are frequently selected, as they are the eight where damage can occur within the patch, and patch 19 is also often selected, as it is close to one of the areas of potential damage in the matrix. However, other patches are sometimes selected, and there may be an advantage in doing so, in order to reduce the numbers of dispersers who might recolonise or boost the populations in recently culled patches. An important aspect of frequently selecting the same patches for culling, though, is that the total number culled has decreased substantially to something nearer 650 to 750 individuals in each replicate.

The improvement in reducing the total damage incurred, as reported in the *SummDamage* file, has increased by something like 3000 units on average. Can we reduce the damage yet further by adopting an even more efficient approach?

If we now change the *Method* to *Random closest to damage X popn size*, we are adding an additional bias towards larger populations in the vicinity of damage locations, and we should find a slight increase in the total numbers culled. However, the decrease in total damage is very limited, if indeed there is a decrease at all.

What about if we adopt a reactive approach, and direct the culling towards patches where the most damage has occurred during the previous year? Select the *Reactive to incurred damage* option for *Method*. This is a deterministic method; the five patches most damaged last year will be selected this year, regardless of their current population or the status of any neighbouring population. The total number culled reduces even more to around 600 to 650 individuals over 30 years, but the total damage incurred increases slightly. This is largely because only those patches in which damage can occur are ever selected; patch 19 is never selected, and therefore damage in the matrix near to patch 19 is never reduced. Patch selection on a reactive basis may therefore not be the ideal solution if damage in the matrix by dispersing individuals is a possibility, but that is not to say that it might not be a good method in other circumstances.

So, on the basis of the scenarios tested so far, it looks like selecting patches randomly, but weighted towards those closest to damage locations, may be the best method for limiting damage by the virtual INNS in this particular landscape. However, we have managed to reduce the total damage over 30 years by only around 25%, which is a start, but not especially commendable. Can we improve on this by altering in some way how we allocate resources each year? For example, we might be able to trade off selecting more patches for culling against a reduced efficiency in each patch, we could change the distance decay coefficient which we have applied, or we could conduct the cull after dispersal rather than before dispersal. Additionally, we might consider how much better we could contain the INNS if we could acquire more resources, e.g. to cull seven patches each year rather than five. RS_CONTAIN does not calculate any possible trade-offs for us, as it has no direct data on the effort required, and we therefore have to simulate a set of plausible alternative scenarios to see which may be best.

We could continue to try different combinations of management control parameters by entering them in the GUI as we have done so far, but there comes a point when this becomes rather tedious (especially if we forget to change the simulation number and overwrite a previous simulation!), and it is better to set up the model in batch mode so that we can run a whole set of alternative scenarios together. Batch mode is also extremely useful for conducting a sensitivity analysis in which we can examine the effects of uncertainty in (some of) the parameters which we have assumed in the model.

6.2 Using batch mode

We will firstly set up in batch mode the no culling (status quo) and best-performing scenarios from the previous tutorial, and check that the results are similar to those we obtained from the GUI, before moving on the compare a range of management control options to try to identify the best way limit the total damage incurred by the INNS.

6.2.1 *Reproducing the GUI models*

You should already be reasonably familiar with the batch files required to specify the demographic and dispersal parameters for this example species from the RS tutorial 4, but in order to allow you to concentrate on setting up the *ManagementFile* for the required culling regimes, all the other batch files are already provided in the *Tutorial2/Inputs* folder.

Edit the skeleton file *ManagementFile.txt* to specify the parameters for the model without management control (section 6.1.1) as simulation 1 and the *Random closest to damage* model (section 6.1.3) as simulation 2. To help you do this, you should use the output parameter files (*Sim*NNN*_Parameters.txt*) from the model you ran using the GUI, and also reference the batch-mode help file *ManagementFile.xlsx* when necessary. Provided that you have the batch executable file *RSbatch_v2_0_CONTAIN_64bit.exe* located in the *Tutorial2* folder, you can simply double-click it to run the model. Any syntax errors will be reported in the output *BatchLog.txt* file, and all errors must be corrected before the model will run.

Check the outputs of this first batch model carefully against the corresponding outputs from the GUI version. In particular, compare the parameter values in the output parameter files, which should match in all respects other than in the names of the landscape and damage map file names and in the habitat codes (the batch version requires habitat codes to be sequential starting from 1, whereas the GUI does not do so, but the two maps are spatially identical). You should also compare the corresponding output *Range*, *Cull*, and *SummDamage* by summarising the data they contain in the same way as you did for the GUI models; the results will not be exactly the same, because RS_CONTAIN runs stochastic models, but they should be closely similar.

6.2.2 *Evaluating more control scenarios*

Once you are completely satisfied that your batch run has reproduced correctly the results you obtained from the GUI version, then it is time to move on to evaluating some new scenarios. Edit *ManagementFile.txt* to add new simulations in which you alter the parameters applied to the *Random closest to damage* culling regime, i.e. columns *Timing* through to *CullBeta*. The *Method* column <u>must be set to 2 for all new simulations</u> (otherwise you will not be comparing variations on the same method). Also, <u>do not change any of the damage columns</u> (*DamageTiming* to *BetaTraversal*); otherwise, you will not be comparing scenarios across the same relationship between incurred damage and local INNS behaviour.

You may add to the batch as many new culling scenarios as you like, but remember that (1) they must be sequentially numbered in the *Simulation* column, and for every new simulation that you add, you must add one (or more) corresponding line(s) in <u>all</u> the other batch input files <u>except</u> the *LandFile*.

When you have run the complete batch successfully, analyse the output files as before to determine which of your alternative scenarios is the most effective at reducing the total damage incurred over the 30 year period.

6.3 An example of an invasive plant species

Although the species in the previous tutorials was an imaginary one, essentially it had the characteristics of a smallish mammal species, both in its demographics and the way its dispersal was modelled, i.e. by SMS. A model for a plant species is rather different in several respects. It is more likely that density dependence will act on development and/or survival than on fecundity, and moreover, the density dependence will be stage-weighted in some way, as large plants tend to influence substantially the vital rates of smaller plants, but not so *vice versa*. Most importantly for the purposes of estimating damage, plant dispersal is more likely to be modelled by a kernel method rather than by SMS, and hence there is therefore no possibility of traversal damage occurring in the matrix. Even if we were using SMS to represent the transfer of seeds by some animal vector that responded to landscape heterogeneity in the matrix, traversal damage is still inappropriate, as the seed does not cause damage during dispersal, but only once it has started to grow, and by definition it cannot grow in the matrix. Thus for a model of a plant INNS, all damage locations should correspond with suitable cells or patches.

6.3.1 Setting up the model

Open the GUI version of the program, *RS_v2_0_CONTAIN_64bit.exe*, and set the working directory to the *Tutorial3* folder.

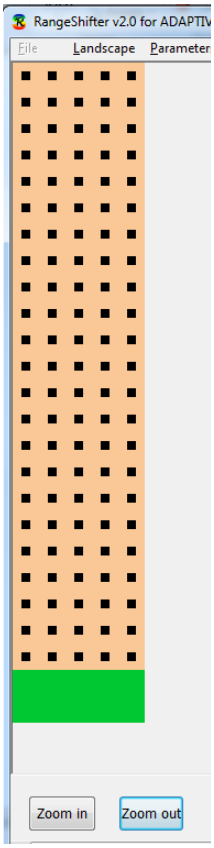

Figure 6.3. Landscape for an invasive tree species

For this exercise, we will run a cell-based model at a resolution of 25 m, and need therefore to load only the habitat file (*Habitat_5x25_25m.txt*) and the damage file (*Damage_5x25_25m.txt*). Once you have loaded these two landscape files, click on the *Zoom out* button several times until you can see the whole landscape on the screen. The landscape measures 5 x 25 cells, and represents a transect leading away from a forest edge (habitat 1, coloured green). There is only one other habitat class present is grassland (habitat 2, coloured beige), all of which is potentially subject to damage by invasion of trees from the forest (Figure 6.3). Note by inspection of the damage file that the maximum possible damage increases with distance from the forest.

Now set up in the *Population dynamics* tab a stage-structured species having five stages and a maximum age of 250 years. We will model a fast-growing hermaphroditic or monoecious tree species in which all mature individuals can produce seed, and therefore we can specify a female-only model.

Set the *1/b* parameters to 200 inds/ha for habitat 1 and to 150 inds/ha for habitat 2.

Now check *Habitat-dependent transition matrices*, and click on the *Set transition matrices* button. That will open a new window, which enables separate transition matrices to be assigned to each habitat class. Type 1 in the *Habitat code* box, click on *Select File* and pick the transition matrix for habitat 1. Then similarly assign the matrix for habitat 2 to that class, and finally click on *OK* to close the window. In this model, the two matrices are very similar, differing only in their fecundity parameters. There are two sapling stages, 1 and 2, and two mature stages, stage 3 having half the fecundity of the fully mature stage 4.

Now uncheck *Density dependence* in *Fecundity* and instead check it for *Survival*. Also check *Survival* for *Stages' weights*, and then click on the *Set weights* button, which will open a new window in which the default weights matrix is displayed. The default matrix has 1.0 in every cell, which means that all individuals are treated equally when calculating the effect of density dependence on local survival in a cell. In order to prevent younger stages affecting the survival of older stages, all entries above the diagonal must be set to zero, but we also require that the mature trees have the strongest influence on the survival of younger stages. Open the file *SurvStgWts.txt* in whatever file browser is installed on your computer, and enter the values in that file into the *Survival* table in the open window. The way that the table is interpreted is the effect of the row stage on the column stage. Click *OK* to close the window and complete the specification of demographic parameters.

Under the *Dispersal* tab, check the *Use full kernel* box, which results in an individual's emigration decision being determined by its dispersal distance drawn from the dispersal kernel rather than as an independent decision. Also check the *Stage dependent* box, and then on the *Sex-/stage-dependent dispersal* tab set *d* to 1.0 for the *juv* stage. This means that all seeds will be dispersed using the default negative exponential kernel of mean 25 m.

Next open the *Management Control* window. We will assume that only stages 2 (large saplings) and 3 (small mature trees) are subject to culling. As this first simulation is to represent the no culling scenario, set the *Threshold density* to 1000 inds/ha and the *Cull rate* to 0.0. Set the *Stage weights* to 1.0 for all five stages. Select the *Logistic* option for *Damage type*, and then change *Alpha* to 0.2, *Beta* to 25.0 and the *Occupancy Option* to *Culled stages density*. Click on the *Draw occupancy* button and check that the incurred damage will be very low below about 10 inds/ha but attain the maximum possible above about 40 inds/ha (Figure 6.4).

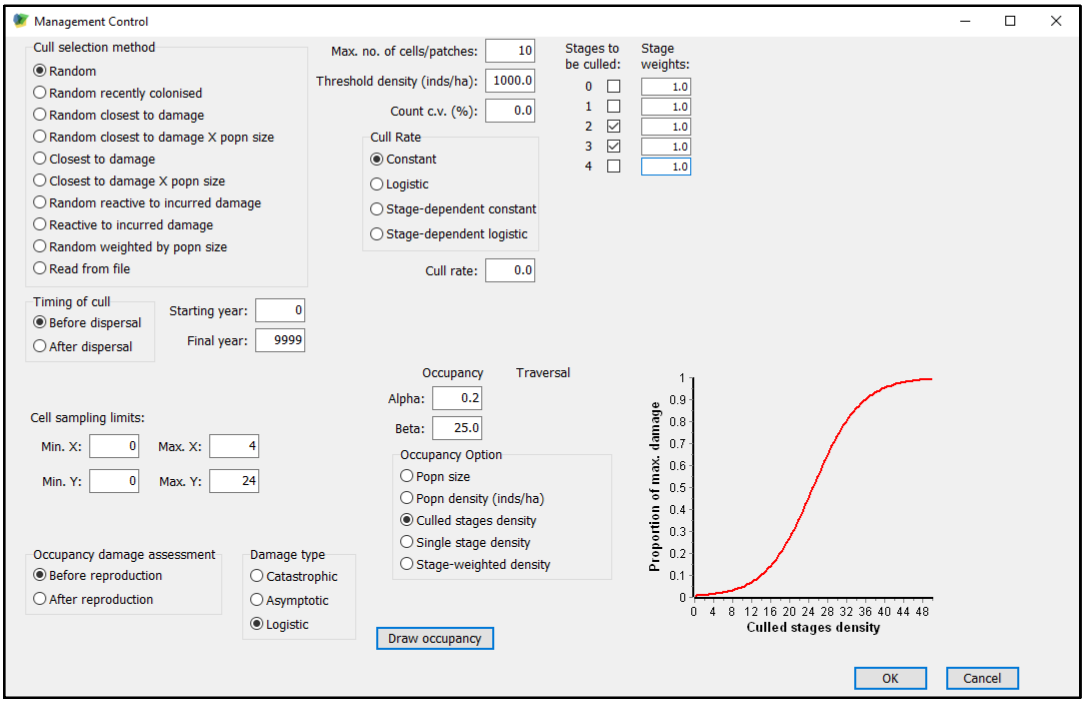

Figure 6.4. Management control settings for the plant species no culling scenario.

Finally set the *Simulation Parameters*. Run 5 replicates for 25 years, and produce *Range* and *Populations* outputs annually. The *Initialisation Rules* should be that there are 10 individuals in each cell, all of which are in stage 4 and are assigned the *Lowest age*. For *Initial Position*, set *Max. Y* to 1 so that only the two southern-most rows (i.e. habitat 1) are initialised.

The model may take a couple of minutes to run, because there are large numbers of seeds to be dispersed each year. Once it has finished, you should see that the number of occupied cells (in RS v2 an occupied cell or patch is defined as one holding a <u>breeding</u> population) remains at 10 for the first 10 years (i.e. the forest cells only), and then starts to increase as invading saplings reach maturity in the grassland. After a further 10 years or so, the occupancy starts to increase again as a second wave of saplings, daughters of the first wave, spread further from the forest edge. The result of this invasion pattern is that the incurred damage also shows three phases, but they start earlier than the occupancy phases, as damage is caused by large saplings as well as mature trees. Note also that the confidence intervals for predicted damage are very narrow; the five replicates predict very similar patterns of damage as the large numbers of invading saplings smooth out across time and space the effects of stochastic variation in dispersal patterns at the individual level.

Add up the total damage incurred over the 25 years - it should be around 10000 units per replicate.

6.3.2 Introducing management control

Now *Refresh* the model and open the *Management Control* window again. Change the *Method* to *Random closest to damage X popn size*, and set the *Distance decay coefficient* to 5.0, which essentially limits the search radius to one cell, so that all cells are assessed only in terms of their own damage risk. We will assume that the grass is fairly short and our field-workers are reasonably competent, and set the *Max. no. of cells* to 10, the *Threshold density* to 0.1 inds/ha and the *Count c.v.* to 5%. Change the *Cull Rate* option to *Logistic*, and set the *Maximum cull rate* to 1, the *Slope* to -0.05 and the *Inflection point* to 100. Draw the cull rate on the screen (Figure 6.5).

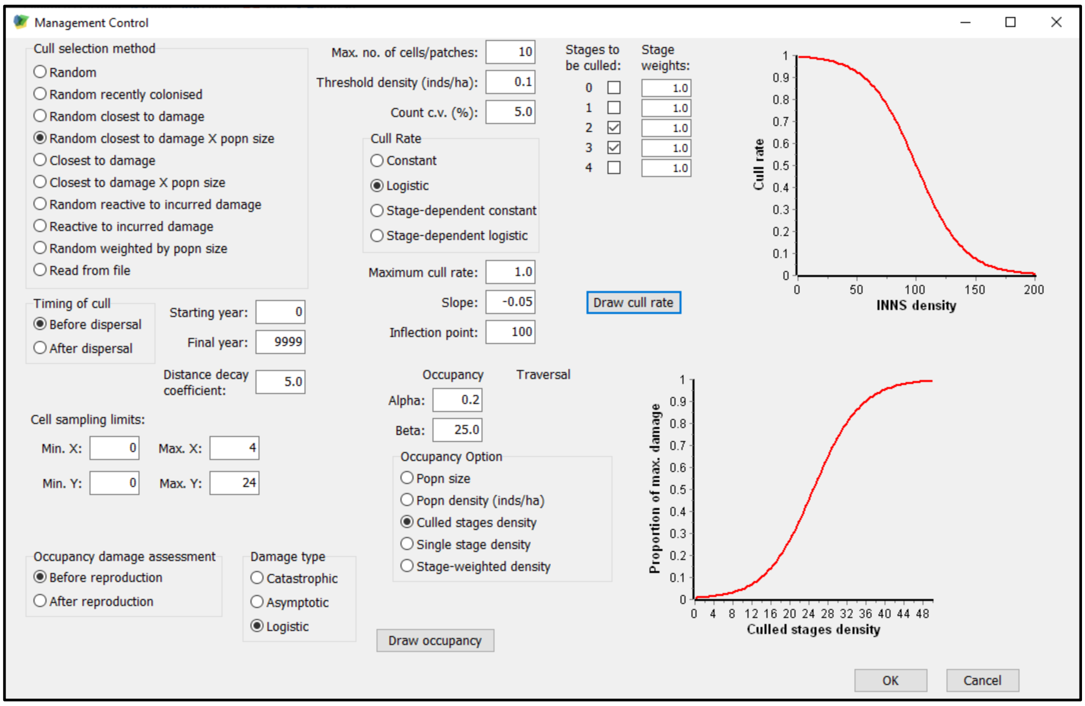

Figure 6.5. Management control settings for the plant species initial culling scenario.

In contrast to the mammal species, for which culling was very inefficient at low density but much higher at higher density, we now have the contrasting relationship: at low sapling density all individuals in a selected cell will be removed, but at high densities they may not be for some reason (e.g. lack of time, the brush-cutter blade breaks, etc.). Of course, it is unlikely that the cull rate will drop to anywhere near zero even for a very high local density; this example is for illustration only, and in reality the inflection point and slope applied should be chosen so that the cull rate decreases to a likely minimum at the highest possible density which could occur.

Once all five replicates have run, you should see that the total damage incurred has been reduced to some extent, and also that the confidence intervals are considerably wider during the third phase from 16 years onwards. This represents uncertainty in our ability to reduce the damage due principally to which cells are culled each year. By culling 10 cells each year, the total damage over 25 years has been reduced by around 35%, but we still see substantial increases in the numbers of smaller mature trees (stage 3) towards the end of the period, and therefore this management control regime of culling 10 cells per year is failing to contain the spread of the INNS. As time goes on, the descendants of these trees will expand the range further and further into the grassland.

6.4 Goal seeking

In the mammal example, we used the batch mode to evaluate which of a possible range of methods might be the most effective, but here we shall use it in a slightly different way. Let us assume that the relationships between cull rate and INNS density and between incurred damage and INNS density are well-established by previous experiments and observations, and therefore we can treat them as fixed. We currently have sufficient government funding to afford to control up to 10 cells per year, but we know that will not be enough to contain the spread of this particular species onto the grassland. We therefore need to make a case to stakeholders for increased funding to control a greater area each year, but we need to know how much more is necessary, and we need to make a good case that it will be sufficient to limit the invasion, even if damage is not completely eradicated.

To do this, we can set up a virtual experiment in batch mode, in which we run a set of simulations across which the number of cells to be culled is increased systematically whilst holding all other parameters constant. We aim to identify a threshold level of control at which the INNS is expected to be contained close to the forest edge.

Edit the *ManagementFile.txt* file in the *Tutorial4* folder to specify the management control options applied in the previous tutorial, bearing in mind that two stages need to be culled. Then run the batch model and check that the input parameters match those applied in the previous tutorial, and that the outputs are closely similar.

Now extend the batch model to run a set of simulations in which you applying increasing levels of the maximum number of cells to be culled. It is up to you exactly what values you apply, but you need to have enough values to be able to identify a threshold level of control (if one exists) whilst limiting the number of simulations to a practical level (since you have also to edit the other batch input files to match the number of simulations). Once you have run the model successfully, use whatever software you like to extract the necessary data from the output *Range* and *SummDamage* files to produce graphs which show how the cell occupancy and the numbers of the damaging stages of the INNS respond to the maximum number of cells culled. The graphs should show appropriate confidence intervals.

7 Notes

A particular requirement, in relation to invasive tree species spreading out from a commercial plantation, is that adult trees invading neighbouring habitats may be culled, but adult trees in the adjacent forest may not be culled. This can in practice be achieved by adding an additional ‘Methuselah’ generation, having the same fecundity as the invading adult trees, having zero mortality and having a minimum age substantially older than any invading adult tree would normally reach (e.g. > simulation period). All the trees in the plantation are initialised as Methuselah trees, and they may not be culled, whereas invading adult trees may be culled outside the plantation.

8 References

Bocedi, G., Palmer, S. C. F., Malchow, A.-K., Zurell, D., Watts, K. & Travis, J. M. J. (2021). RangeShifter 2.0: An extended and enhanced platform for modelling spatial eco-evolutionary dynamics and species’ responses to environmental changes. *Ecography*, **44**, 1453-1462.

Caplat, P., Nathan, R. & Buckley, Y. M. (2012). Seed terminal velocity, wind turbulence, and demography drive the spread of an invasive tree in an analytical model. *Ecology*, **93**, 368-377.

Clark, J. S., Silman, M., Kern, R., Macklin, E. & HilleRisLambers, J. (1999). Seed dispersal near and far: patterns across temperate and tropical forests. *Ecology*, **80**, 1475-1494.

Powell, P. A. & Aráoz, E. (2018). Biological and environmental effects on fine-scale seed dispersal of an invasive tree in a secondary subtropical forest. *Biological Invasions*, **20**, 461-473.

Travis, J. M. J., Harris, C. M., Park, K. J. & Bullock, J. M. (2011). Improving prediction and management of range expansions by combining analytical and individual-based modelling approaches. *Methods in Ecology and Evolution*, **2**, 477-488.