Forest Roads Simulation v1.0
LANDIS-II Extension
User Guide

Clément Hardy[1]

[1]Université du Québec à Montréal

Last Revised: October 11, 2020

Table of Contents

1 Introduction

This document describes the Forest Roads Simulation or “FRS” module (‘plug-in’) for the LANDIS-II model. Users should read the *LANDIS-II Model User’s Guide* prior to reading this document.

1.1 A warning about the terminology used in this guide

In this guide, be warned that the terms “pixel”, “cell” or “site” are used interchangeably. This is due to the fact that the LANDIS-II model operates on a space that is discretized into “sites” or “cells”, which are units of space that are distributed in a two-dimensional matrix. The state of these cells/sites, or values that they contain can thus be represented by pixels, making a raster image. Therefore, while we use those three terms throughout the guide, they can be seen as meaning the same thing.

Additionally, we use the terms “forest roads” and “roads” in two distinct ways. This is because the FRS module is supposed to create forest roads, but not main, paved roads that are consider as unchanging throughout the simulation. Therefore, we use the term “forest roads” to refer to the roads that the model can create or destroy; and when we only say “roads”, we mean both forest roads and the main paved roads of the simulated landscape.

1.2 The road Landscape

The Forest Roads Simulation module deals with the evolution of forest roads in the LANDIS-II landscape that are used to access zones to harvest. These zones are chosen and harvested by another LANDIS-II extension that manages harvesting disturbances (e.g. “Base Harvest” or “Biomass Harvest”). Therefore, **the FRS module works in tandem with a harvest module**.

To function, the FRS module edits a landscape parallel to the main landscape used in the LANDIS-II simulation, except that it contains only forest roads. This landscape will be outputted by the extension as a raster that the user can easily display when the simulation is completed.

1.2.1 Sites without roads

For each LANDIS-II site, the “road landscape” contains a value that describes if the site contains a road or not. Sites without roads have no value, or a value of “0” in the road landscape. These sites can be in a disturbed state (harvested or burned) or could be seen as contain skidding tracks (if they were harvested with an already existing road nearby), but they do not contain a proper road.

1.2.2 Sites with exit points (e.g. sawmills or with pieces of the main paved road network)

Another category of values for the sites in the road landscape concerns the presence of sawmills or paved roads that are part of the main road network of the landscape. This category is considered by the module as places where the wood harvested in the landscape can "exit" the landscape (to be sold or used). Every initial road landscape must have at least one site of this category, as the algorithm will try to connect newly harvested areas to an exit point directly, or through pixels with forest roads (see below). Currently, exit points are static throughout the simulation: they are not created or deleted through time.

1.2.3 Sites with forest roads

Finally, the sites can be indicated as containing a forest road used to transport timber from the harvested sites to an exit point (e.g. sawmills, or the main paved road network; see above). These forest roads can be of different types. The types of roads are entered by the user in the parameter file, allowing flexibility in the names and categories used. The type of the forest road present on a given site at the beginning of the simulation is indicated by the user in the raster input file for the road network; then, it can be modified as the network evolves with the aging of the roads and an algorithm that models the flux of the harvested wood through the road landscape (see below).

1.3 Determining the type of forest roads

Forest roads can be described as different categories, relative to their dimensions, their materials, and so on. In reality, we expect that forest roads of higher categories are only constructed to facilitate the transport of timber as they are costly to build, even if no clear guideline exists on this subject to our best knowledge. Therefore, to reproduce the evolution of a forest road network in a way that is both realistic and easy to understand, we used an approach that inspires from the study of hydrological networks.

In this approach, the forest road network is perceived as a watershed, with the "water" being timber, and coming from the extremities of the forest road network (the harvesting sites). However, while water follows the slope of the land to flow through the watershed, the timber transport will follow the least cost path to the closest place where the wood can exit the landscape.

Thus, to determine the type of each forest road and how the network evolves, the "flux" of timber is simply calculated for each segment of the forest road network by addition from segment to segment. The user can then indicate thresholds of timber flux by time step for which a certain forest type must be built to allow for transportation of this flux.

1.4 Calculating the tracing of the forest road with the least-cost path and the cost of construction

To compute the tracing of the forest roads in the landscape (that are created throughout the simulation), the FRS module is based on the hypothesis that the right tracing for a given forest road is going to be the cheapest one, as forest roads represent an important cost associated with forest management.

To this end, a great number of methods exist in order to determine the least-cost path between two given points. The resolution of this problem is called, in the context of forestry, the "Single Target Access Problem" (STAP; Chung et al., 2008). It is quite simple to resolve. However, the design of a forest road network requires us to deal with a "Multiple Target Access Problem" (MTAP), for which optimization of a solution is much harder to deal with (Chung et al., 2008).

To be able to give a relatively "good" solution to a MTAP, current software for designing forest road networks such as REMSOFT's Road Optimizer (Remsoft, 2019) uses a heuristic coupled with a path-finding algorithm to solve the MTAP as a sequence of STAPs ordered via the heuristic. For example, the heuristic can be "deal with the farthest harvested site from current roads first"; and the path-finding algorithm can be one as simple as the Dijkstra algorithm. Therefore, with this approach, a least-cost path will be found for each harvest site by the Dijkstra algorithm one by one, in the order of their distance to existing roads. The heuristic chosen will thus heavily influence the shape of the forest road network; and the solution proposed by such a method, while not optimal, is easily interpreted as multiple heuristics can be used successively to identify differences or similarities between the generated networks.

We employ the same approach in this extension. Thus, the least-cost path between two points is currently determined by the Dijkstra algorithm, and three heuristics are proposed to the user: Random, closest first, farthest first.

To determine the cost of moving from a site to another, the extension can use up to five sources of information that are important for creating a forest road: distance, slope (elevation), existing bodies of water (rivers, lakes), and soils. While it would be impossible to propose an operational cost of construction for forest roads with the discrete resolution used by LANDIS-II, a global idea of the cost of a forest road can be computed using these values.

The cost of the construction of a forest road in the model is thus influenced by two things: the addition of the cost of transition from one site to another (using the five information about the landscape described previously); and the type of the forest road determined by the “timber flux” algorithm (see previous sub-section).

1.5 Major Releases

1.5.1 Version 1.0 (October 2020)

Features:

- All of the essential functions of the model present
- Code cleaned up

1.6 Minor Releases

1.6.1 Version 0.5 (August 2019)

First pre-release for testing purpose. Features:

- Detection of an installed harvest extension (without it, the road network will not change).
- Reading of a raster containing the initial road network.
- Completion of the initial road network to avoid isolated roads that lead to nowhere.
- At each time step, the extension gets all of the recently harvest sites, and construct a forest road to them unless they are close enough for to an existing road for skipping wood to them.
- An output raster is created at each time step to see the evolution of the road network in the landscape.

1.6.2 Version 0.9 (October 2019)

Pre-release for research purposes. Features:

- The trajectories of the forest roads can now be influenced by zones where roads are forbidden, existing roads, elevation, fine topography (e.g. cliffs), water, streams and soil types. The extension can accommodate only a part of these information, according to what the data that the user can use.
- The user can now enter customized types of roads in the parameter file.
- The flux of wood through the forest roads can be simulated. If so, forest roads can be upgraded to accommodate the flux going through them. The threshold above which the forest roads should be upgraded are defined by the user.
- If the flux of wood is simulated, the extension will output rasters picturing the flux through the landscape for a given time step.
- The aging of the forest roads can be simulated. After a certain age inputted by the user in the parameter file, a forest road is considered to be destroyed, and must be rebuilt or maintained in order for wood to go through it again. A forest road upgrade or construction/maintenance resets the age of a forest road.
- A log saves different information related to roads for each time step.

1.6.3 Version 0.95 (November 2019)

Pre-release for research purposes. Features:

- Huge improvements to performances and simplification of the code.

1.6.4 Version 0.96 (January 2020)

Pre-release for research purposes. Features:

- Updated functions for getting the “TimeSinceLastEvent” variable for harvest events.
- Performance improvements.

1.6.5 Version 0.97 (June 2020)

Pre-release for research purposes. Features:

- Looping behavior: The module can now create loops in the forest road network, in order to account for more realistic fragmentation measures
- The module now takes into account the fact that if a repeated harvest (single or multiple repeat) is used on a cell, then a forest road will be needed to come back at the next rotation. Therefore, if road aging is taken into account in the parameter file of the module, the module will then try to find the best (cheapest) option between creating a lower road type that will get destroyed before the next rotation (due to road aging) and that will need to be re-constructed to access the harvested site again; or building a more solid, but more costly type of forest road that will last until the next harvest rotation.
- Fixed a bug concerning the wood fluxes that resulted in some incoherent forest road types.

1.6.6 Version 0.98 (June 2020)

Pre-release for research purposes. Features:

- Fix a bug during the initialization phase that was due to the module trying to gather non-initialized variables at that time.

1.7 References

Chung, W., Stückelberger, J., Aruga, K. et Cundy, T. W. (2008). Forest road network design using a trade-off analysis between skidding and road construction costs. *Canadian Journal of Forest Research*, *38* (3), 439-448. doi: 10.1139/X07-170

Remsoft. (2019). *Remsoft Road Optimizer*. https://www.remsoft.com/road-optimizer/

1.8 Acknowledgments

Funding for my thesis, that allowed the development of this extension, is due to the Natural Sciences and Engineering Research Council of Canada.

I deeply thank the LANDIS-II foundation for their incredible work toward providing an easy to use, and easy to program model.

2 Other input parameters

These basic parameters should be at the beginning of the parameter .txt file (see section 6 for an example)

2.1 LandisData

This parameter's value must be `" Forest Roads Simulation"`.

2.2 Time step

This parameter is the extension's time step. To avoid errors, it should be the same as the harvest extension. Value: integer > 0. Units: years.

2.3 Heuristic for network construction

This parameter determine which heuristic will be used to solve the MTAP problem that represents the construction of a forest road network (see sub-section 1.4). This heuristic allows the extension to find a solution concerning the shape of the forest road network. However, this solution will be sub-optimal; but as optimizing a MTAP problem is extremely difficult and requires an enormous amount of calculations, this simpler method, used in existing professional software, is used by the module instead.

The parameter "HeuristicForNetworkConstruction" can thus take three different values: "Random", "ClosestFirst", "FarthestFirst". With the "Random" value, at each time step, the recently harvested sites will be connected to the road network by the Dijkstra algorithm (which finds a least-cost path between two points) one by one, and in a random order. With the "ClosestFirst" value, the sites closest to existing roads will connected first. The opposite will happen with the "FarthestFirst" value.

We recommend the user to try different simulations with different heuristics to determine if this affects their simulated landscape or not. However, previous testing from our part seem to have shown that the "ClosestFirst" heuristic provided the most realistic results when compared to existing forest road networks.

2.4 Skidding distance

The skidding distance indicates the distance above which it becomes necessary to construct a forest road to access a site that is harvested. Thus, if the harvested site is at a euclidean distance from the closet site with a road on it that is lower than this parameter, then no forest road will be constructed to this harvested site. It is supposed that the wood will be transported by skidding instead in such a case. In any other case, a road will be constructed to this site.

2.5 Looping behavior

As it is currently formulated, our FRS module creates a road network that resembles the branches of a tree, with the branches never touching each other. While such a structure can imitate the density of roads existing in reality in a landscape, or the paths that they take globally, it can really influence the fragmentation of the landscape.

Indeed, this can be an issue for some fragmentation indices that are based on the existence of patches of habitat, if those patches are considered as one as long as they represent a spatially continuous mass of pixels of the same type. In that situation, one can easily visualize that a road network looking like a tree with branches that do not touch each over will not surround habitat pixels completely, which can leads to big patches of habitat existing despite very road densities.

To fix this issue, we introduced the possibility of creating “loops” in the network, meaning places when the branches of the network connect to each other, and thus become able to completely surround habitat. We call the part of the module that is responsible for this the “looping algorithm”.

To activate the looping algorithm, the user must associate the value “yes” to the parameter “LoopingBehavior”. If that's the case, 3 other parameters ruling the functioning of the algorithm must be given (see sub-sections below). To avoid using the looping algorithm, the value "no” must be associated to the “LoopingBehavior” parameter. If that’s the case, no other parameter needs to be given in addition to this one.

2.5.1 Looping distance

The “LoopingDistance” parameter must be given if the looping algorithm is used. It correspond to a euclidean distance. When a harvested site is considered to connect it to an existing forest road or to an exit point via the Dijkstra path-finding algorithm (see section 1.4), the looping algorithm will check if forest roads or exit points already exist at a distance corresponding to the “LoopingDistance” parameter. If that is not the case, the looping algorithm will not create a loop, in order to avoid loops that are made over very long distances. Instead, it will only construct one road, as if the looping algorithm was not activated. If that was not the case, loops would form almost everywhere, leading to an over-crowded road network; in addition, we expect that when forest roads are created by engineers, loops are only considered when they do not represent an additional cost that is too big.

Additionally, the looping distance will also be used to force the algorithm to create a loop connecting two existing forest roads or exit points to the considered harvested site that are as far away from each other as possible. This is done as the looping algorithm will create a first road from the harvested site, then a second one. But when the first road is completed, the looping algorithm will define all existing forest roads and exit points at a looping distance from the one that was reached by the first road as forbidden for the construction for the second road, forcing the Dijkstra path-finding algorithm to look further in order to complete the second road.

Thus, the “LoopingDistance” parameter should be any value superior to 0.

2.5.2 Looping Maximum Percentage of Roads

Once that the looping algorithm have checked if there are roads nearby (see previous section), it will check if there are not too many roads. If roads are too numerous in the range of the looping distance (see previous section), the looping algorithm will not build a loop. Instead, only one road will be constructed, as if the looping algorithm was not activated.

This additional precaution is present in order to avoid creating loop in a part of the landscape that is already over-crowded with roads, and maybe loops that have been created before. Thus, the “LoopingMaxPercentageOfRoads” represents a percentage of cells at a looping distance (see previous section) from the considered harvested site that already have a road on them. If this percentage is reached or surpassed, no loop will be created.

Thus, the “LoopingMaxPercentageOfRoads” parameter should be any value between 0 and 100.

2.5.3 Looping Maximum Cost

Once that the looping algorithm have checked if there are roads nearby, but not too much (see previous sections), it will attempt to create a first road from the considered harvested site to the nearest existing forest road or exit point using the Dijkstra path-finding algorithm. Then, it will attempt to create a second road from the considered harvested site to reach another existing road cell, with the one being reached by the first road and the roads surrounding it being forbidden, in order to force the algorithm to link two roads that are as further away from each over as possible (see section 2.5.1).

When the path of this second is determined by the Dijkstra path-finding algorithm, the looping algorithm will check the cost of construction of this second road, and will compare it to the cost of construction of the first road. If the cost of construction of the second road is too high, then only the first road will be constructed, which will result in the same behavior as if the looping algorithm wasn't activated.

This precaution is done in order to prevent the looping algorithm from constructing very costly roads in order to create a loop (e.g. crossing a body of water, constructing a very expensive bridge as a consequence). The parameter controlling the comparison between the cost of the first road and the cost of the second road is the "LoopingMaxCost" parameter. It represents the maximum value that the division of the cost of the second road by the cost of the first road can reach for the second road to be constructed.

As an example: if the first road costed 200 000$ to construct, and the second road costed 1 000 000$ to construct, with the "LoopingMaxCost" parameter being "3", then the second road will not be constructed, as its cost of construction is more than 3 times the cost of the first road. If its cost of construction is under 600 000$ however, then the second road will be constructed.

Hence, the "LoopingMaxCost" parameter must be a value superior to 0.

2.6 Location of the outputs rasters

This parameter indicates where the user wants the output rasters generated at each time step to be saved. It can be expressed as a relative or an absolute path. The name of the output rasters must be indicated, including the ".tif" extension (e.g. "./output/roadsRasters/roadNetwork.tif"). The Forest Roads Simulation extension will then automatically add the number of the time step at which the output raster was made at the end of the raster's name.

2.7 Location of the summary log of the costs of construction

This parameter indicates where the user wants the summary log of the costs of construction to be located. This summary log will contain the cost of construction of all of the forest roads constructed during the simulation, the number of roads constructed at each time step, and other useful information.

3 The input rasters

To function, the Forest Roads Simulation module requires another set of spatial data than the data already contained inside LANDIS-II. This spatial data is fed to the extension by the use of different input rasters. These rasters, in turn, have to be in a certain format to be read and understood by the module. In this section, you will find the details and some indication of how to generate those rasters easily. More examples can be found in the article of Hardy *et al*. (2020), currently in process of publication.

The FRS module cannot work without an initial road network raster containing at least one pixel/site being an exit point for the wood to go (see below). It also requires a raster indicating the zones where roads can be built in the landscape. With these two obligatory rasters put aside, every other input raster is optional. Yet, without them, the path of the forest roads that the module creates might not be realistic, as they each represent an important variable to be taken into account during the path-finding algorithm.

Thus, if a certain raster is not fed to the extension (except for the initial road network raster), the extension will just a warning to the user, and will not use the information it would have contained during the path-finding process.

Each one of the input rasters must have the same resolution and extent as the other LANDIS-II rasters. If not, the extension will throw an error to the user.

3.1 The raster of buildable zones

This raster simply indicates to the extension where forest roads can or cannot be built in the landscape. Any pixel with a value different of 0 is considered “buildable”. Consequently, any pixel with a value of 0 is considered “non-buildable”, and no forest road will be built on the cell/site corresponding to the pixel during the simulation.

3.2 The initial road network raster

At the beginning of the simulation, the initial road network is loaded from a raster given by the user. This raster must respect certain criteria to be accepted by the extension.

As any LANDIS-II raster, it must be formatted as integers. The values to enter are relative to the road IDs given by the users in the parameter files. Values equal or inferior to 0 are reserved for sites without roads on them, while positive values should each correspond to a road ID present in the road types described by the user (see sections 5.3.3 and 5.4 below). If a value for a pixel is neither negative, nor 0, nor one of the positive value indicated as the ID of a road type, the FRS module will return an error when he'll try to access this road, as the road type will be unknown. Here is an example of values for this raster (but remember that positive values will depend on the user's choices):

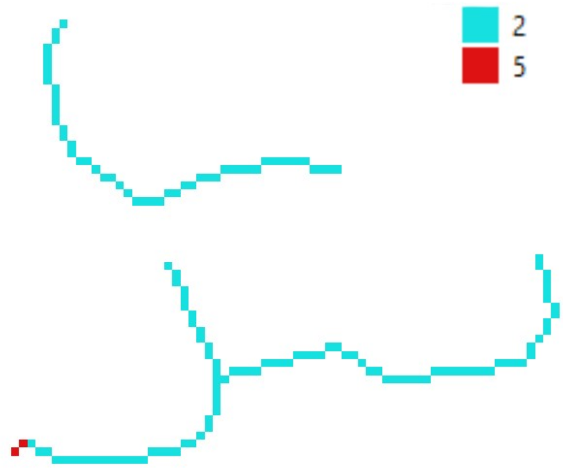
Raster pixel value	Road type
>= 0	No road
1	Primary forest road
2	Secondary forest road
3	Tertiary forest road
4	Winter/temporary forest road
5	Sawmill
6	Main road network (paved)
Any other value	Not valid (unknown)

The same values will be used in the output rasters containing the road network generated by the extension.

To create the initial road network raster, you can use the free GIS software "Quantum GIS" (QGIS). You only need to load a vectorial layer containing the road network (forest roads included) of your landscape, and to open the attribute table of your layer to create a new field. You can then use the "field calculator" of QGIS to transform another attribute field that contains information about your road type into the corresponding integer values (e.g. 1, 2, 3, 4, 5, or 6 in the above example).

Once this new field is created, you can simply use the "Vector to Raster" tool in QGIS to "burn" the attribute field you just created (containing the integer values) with the right resolution and extent. At this point, it is recommended that you define the "No Data Value" of the raster at 0, and that you use the advanced parameter to save the raster's values as "int16".

Here is a very simple example raster following such instructions:



As you can see, this example raster only contains two pixels/sites indicated (in the example classification displayed above) as a sawmill (5), and pixels/sites indicated as secondary forest roads (2). The rest will be sites without roads (0 or No Data).

You must remember that this raster must have at least one pixel/site that is of the “Sawmill” or “Main road network” category. If not, the extension will throw an error, stopping the simulation.

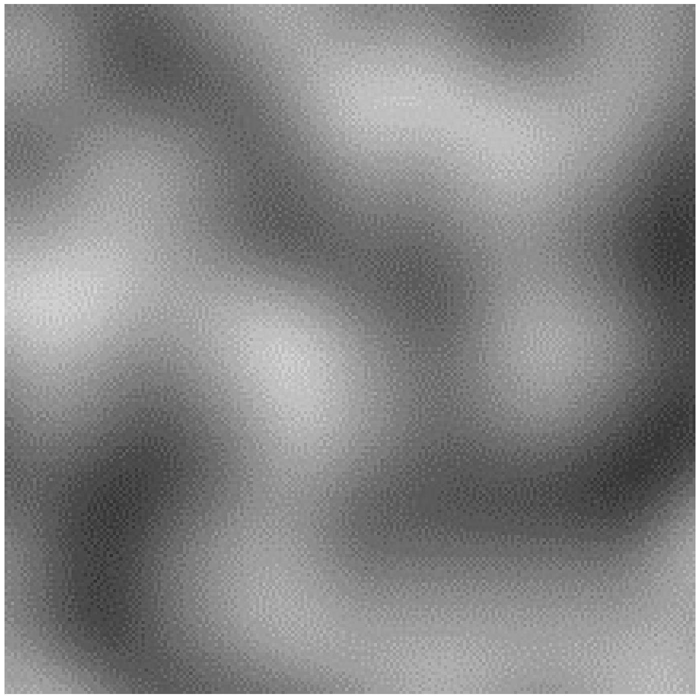
3.3 The coarse elevation raster

One of the main parameters to take into account during the construction of forest roads is the slope of the land, as it is always more complex for the same type of terrain to build a forest road on a high slope than on a flat area.

However, because the resolution of LANDIS-II too low to take into account small changes in slope, we separated the topography of the landscape into two rasters of different resolution, containing two distinct (albeit related) information. The first is the “coarse” elevation raster.

In this raster, each pixel should contains the value of the mean altitude associated with the pixel/site. To create such a raster, we recommend the use of a Digital Elevation Model (DEM) such as LiDAR data, and to change the resolution and the extent of the DEM to the one used for your LANDIS-II rasters. The elevation should be in meters to adapt to the formula used in the FRS module to compute the slope between two pixels.

When displayed, the coarse elevation raster should look like this:



3.4 The fine elevation raster

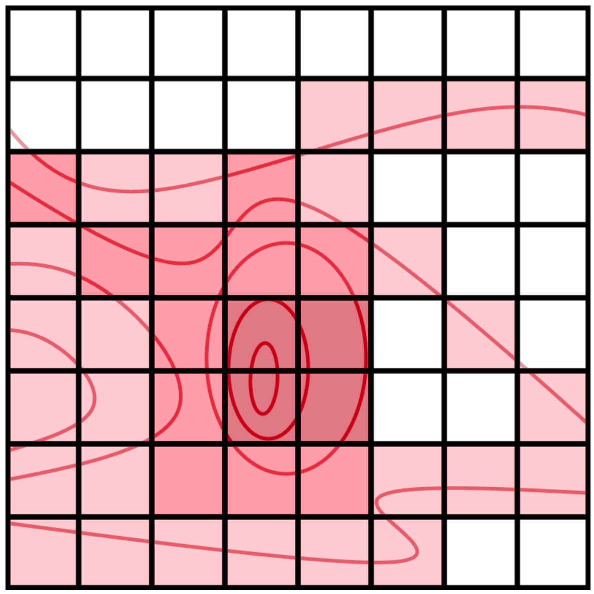
The second elevation raster will contain information pertaining to small changes in the topography that might become an obstacle for the construction of the forest road, requiring detours or an increase of road distance to traverse the pixel/site (e.g. cliffs, breaks, ditches, etc.).

Ideally, this raster should contain value related to the quantity and closeness of contour lines in the cell. This can be done in several way. We propose what we believe is the simplest one, but others can be attempted.

To create such a raster, the user can thus use vectorial data of contour lines. These contour lines can be derived from a Digital Elevation Model using a software like QGIS; or, it can be found directly from other sources. Once the contour lines are obtained, the user can input the number or length of the lines crossing a given pixel/site into a raster.

In QGIS, this can be done by using 3 tools: first, the “grid” tool to create a polygonal square grid with the same extent/resolution than the desired raster. Secondly, the “Sum line length” tool to fill an attribute field of this vectorial grid with the length or quantity of the contours lines that cross each square polygon of the grid. Thirdly, the “Raster to Vector” tool to “burn” the attribute field created previously into a raster with the correct resolution, extent and value format (int16; see sub-section 3.1). Every value can be used, but should not reach above the int16 limit (32,768). We recommend that a measure in meters should be used, to keep a uniformity among the values used by the module.

Here is a representation of what the fine elevation raster looks like, with the red lines corresponding to contour lines, and redder pixels representing higher values in the raster:



3.5 The coarse water raster

Two rasters that concerns bodies of water will indicate another costly obstacle for the construction of forest roads: the presence of rivers, lakes or streams.

As concerning elevation, the presence of water is separated into two rasters. The first water raster will concern large bodies of water, such as rivers and lakes. To create such a raster, the user can use polygons showing the extent of rivers and lakes in the landscape. Using simple tools, he can put the following values in the pixels of the raster.

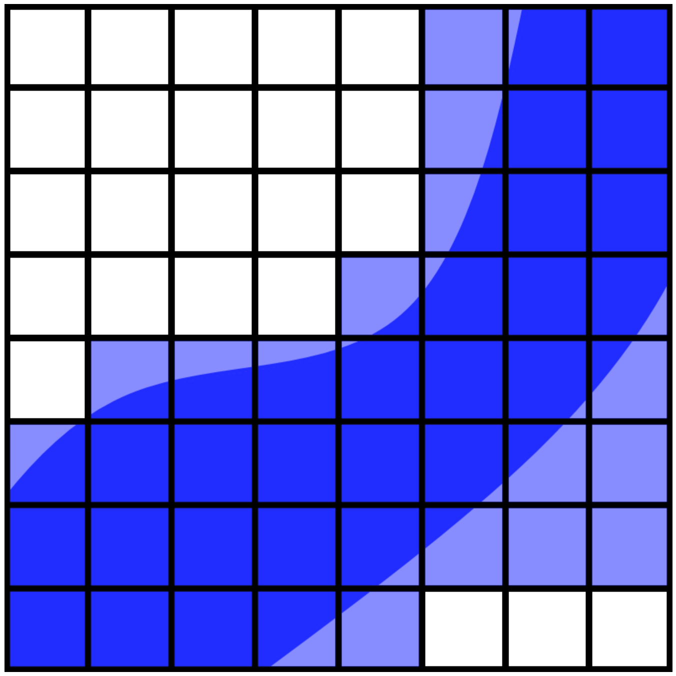
Users should remain careful about small rivers (smaller in width than the pixel size) which might create “gaps” during the rasterization process. Indeed, polygons that do not cover the entire area of a superposing pixel are not inserted in the given pixel according to some GIS tools for rasterization. These gaps, in return, can be wrongly used by the path-finding algorithm to traverse the rivers.

To avoid this, we recommend the use of the “polygons to lines” tool of QGIS to first transform the polygons representing the bodies of water into lines (the lines being their contour). Then, a first raster will be created by using the tool “rasterization” of QGIS on the layer of line created before, that will create a raster where every pixel in contact with one of those lines will be associated with the correct value. Then, the “rasterization” tool can be used again of the polygon layer to create a second raster where only pixels entirely covered by a polygon representing a body of water will be associated with the correct value. Then, the “merge” tool for rasters of QGIS can be used to merge those two rasters, resulting in a correct raster with no gaps.

Here is a table representing a possible set of values that will be found in the coarse water raster. These can change, as the FRS module will simply consider pixels inferior of equal to 0 as not containing a body of water, and every positive raster as containing a body of water. Currently, the FRS module does not distinguish between different types of bodies of water.

Raster pixel value	Body of water
>= 0	No body of water
1	River
2	Lake

Here is a representation of what the coarse water raster should look like, with the dark blue shape being a polygon corresponding to a body of water, and light squares representing pixels with a value superior to 0 in the resulting coarse water raster:

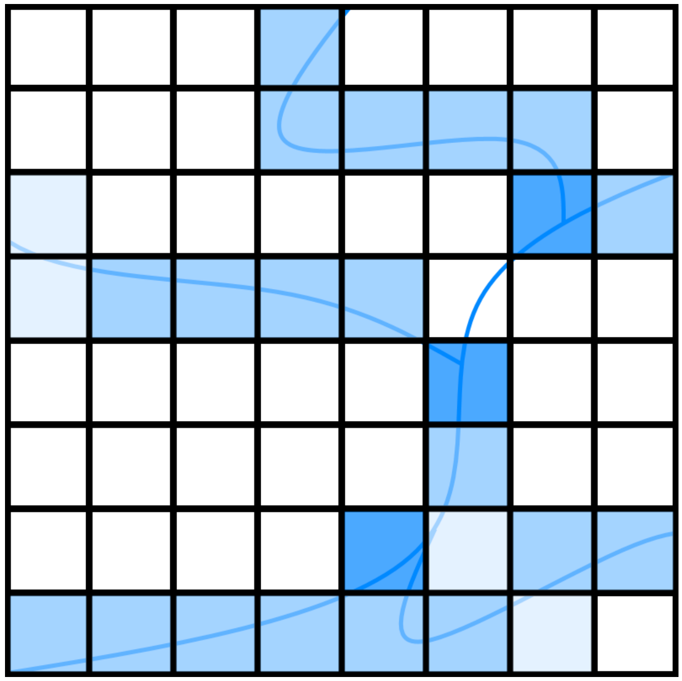


3.6 The fine water raster

This second water raster will contain information about streams. We recommend to create it with the same tools as those advised for the fine elevation raster (see section 3.4), but this time by using polylines that describe the position of streams in the landscape.

Thus, the corresponding raster should contain, in each pixel, a measure of the length of streams present in that pixel. This length can be from only one stream or several; it does not matter. What matters is that as the length of streams in the pixel/site increase, the probability of having to cross at least one increase too, indicating that one or multiple culverts will have to be installed to cross it, leading to increased cost of constructions.

Here is a representation of what the fine water raster should look like, with blue lines representing streams, and bluer pixels representing higher values in the raster:



3.7 The soil raster

This last raster will give information to the FRS module concerning the soils of the landscape. This can be important for the construction of forest roads, especially if the soils allow the builders and engineers to use local material (gravel, etc.) to build the forest road; or if materials have to be transported to the location, which can greatly improve the cost of construction.

In each pixel of this raster, there should be an integer corresponding to an additional cost related to the type of soil present in the pixel. This additional cost can be determined using a linear model applied on construction costs of real forest roads. If done so, the intercept of the model will contain the effect of a "reference" soil type; we recommend that this "reference" soil type is the type of soil for which the cost of construction is the lowest. Hence, every over soil type will be associated to a positive additional cost. The intercept of such a linear model will thus become the distance cost parameter (see section 4 bellow more detailed examples).

4 The cost parameters

Following the location of each of the input rasters must be a set of parameters that indicates the cost associated with the variables of the rasters for the path-finding algorithm. We recommend that all of the costs represent real values in the monetary unit chosen by the user, so that both the input and the output of the extension are grounded in values that are easy to discuss or relate too, making the parametrization and the interpretation of results easier with both modelers and non-modelers.

These costs will be expressed for one chosen type of forest road category, like a primary, secondary or tertiary forest road. Then, a multiplication parameter will be applied when forest roads of lower or higher rank will be constructed to increase or reduce the cost (see section 5.3.4).

4.1 Distance cost

This parameter must follow the location of the initial network raster, and is the only obligatory parameter to be given in the parameter file. This parameter will represent the irreducible cost of building a forest road of a chosen type on the distance of a side of a site. For example, if the sites of the LANDIS-II landscape have 100m of side, then this cost must be the irreducible cost of building 100m of forest road on the best conditions possibles for forest road construction (e.g. 0% slope, the best soil type, etc).

If the user chooses to not input other costs rasters, the algorithm will only base the determination of the least-cost path on this cost. Therefore, if no over input raster is indicated, this cost can contain a mean of other costs (crossing streams; carrying materials for the structure of the forest road; taking topography into account; etc.) in order for the cost outputs to be more realistic. However, the trajectories of the forest roads created by the algorithm will be entirely unrealistic, as the algorithm will only try to reduce the number of cells used for the forest road (as each cell will be associated with the same cost). It is thus highly recommended that at least the coarse elevation cost (see section 3.3) and the coarse elevation costs (see section 4.2) be imputed.

This parameter can easily be estimated using a linear model applied to real construction cost of forest roads. If done so, the basic distance cost will be the intercept, free of any of the effect of other factors (slope, soil type, etc.), or containing only the reference level of those factors (e.g. 0% slope, best soil type for construction, etc.).

The parameter should thus be a single value superior to 0.

4.2 Coarse elevation costs

These costs correspond to costs added to the basic distance cost due to the necessity of constructing a forest road on a slope. The user can input an additional cost for several ranges of slope percentage. The slope percentage for a pixel will be equal to the biggest slope percentage among its 8 neighboring pixels, and is calculated using the pixel size indicated to LANDIS-II and the coarse elevation raster.

The costs take the form of a table indicating a lower threshold and an upper threshold in percentage of slope, and the associated cost for values slope percentages between those two thresholds. Thresholds must be continuous and encompass all of the possible slope values in the landscape. An example is given in the example parameter file (see section 6).

The cost associated to each slope range should be a single value superior to 0. If no coarse elevation raster has been given (the parameter has been set as “none”), then those costs will not be taken into account during the path-finding process.

4.3 Fine elevation cost

These costs corresponds to a multiplication of the cost of construction of the forest road on the cell due to the fact that the road will have to slalom in order to avoid fine topographic elements (cliffs, depressions, etc.). In return, this will increase the distance of forest road that will have to be constructed in order to pass through the given site.

The parameter will be written as a table of values with three columns: Lower topography threshold, upper topography threshold, and multiplication value for this range of values. The thresholds should correspond to thresholds in the values of the fine elevation raster; the associated multiplication will then be applied to the cost for crossing the cell (that already contains the effect of all of the other costs), according to the fine topography values of the cell.

The thresholds should thus be values ranging between 0 and the maximum topography value existing in the fine elevation raster; the first lower threshold must be smaller than the smallest fine elevation value in the raster (we recommend to use “0”); and the multiplication should be a single value superior to 0.

4.4 Coarse water cost

This cost should simply be the cost of building the section of a bridge on a body of water equal to the length of the site where it is present. This allows the algorithm to cross lakes and rivers, but at a great cost. For example, if the price of a basic bridge over 100m of water is around 2 000 000$, and if the side of a site in the LANDIS-II landscape is of 50m, then this cost should be of 1 000 000$. While the price of bridges can vary according to the type of bridge and the material used, the coarse water cost is mostly used to discourage the path-finding algorithm from crossing large bodies of water.

The parameter should thus be a single value superior to 0.

4.5 Fine water cost

This cost is the cost associated with the crossing of a stream, which will necessitate the construction of a culvert. Therefore, we recommend to use the mean cost of installation of a culvert as a value for this cost. The number of culverts that will have to be created by crossing the given site will be dependent on the probability of having to cross a stream; which, in turn, depend on the length of streams present in the given site, as indicated by the fine water raster.

Thus, the number of culverts to be created by crossing the site will be the result of the division between the length of streams in the site and the length of the diagonal of the site. For example, if the site has a diagonal of 140m, and if 300m of streams are present in the site; then given the probability of crossing the streams, 2 culverts to install will have to be taken into account by the cost algorithm.

The parameter should thus be a single value superior to 0.

5 Road type thresholds and cost multiplication values

As indicated in sub-section 1.3, the type of the forest roads to construct can be deduced by the use of an algorithm that considers the forest road network as a kind of watershed. In this view, each recently harvested site becomes a source of wood at the considered time step. The amount of wood coming from a site is currently measured by the number of age cohorts harvested in the site.

Once all of the forest roads have been created at the current time step to access the recently harvested sites, the algorithm will then determine if any forest road needs to be upgraded to accommodate for the flux of wood going through the road network at this time step.

To do this, the algorithm will make the timber from the harvested sites “flow” from road cell to road cell, until it reaches an exit point. In consequence, the type of the forest roads (which are always constructed from the “lowest”, cheapest road type) will be upgraded to the one determined by the road type thresholds (see section 5.3.1 and 5.3.2). Hence, for a given forest road, if the additive flux of timber that goes through this road at the current time step is between the thresholds of a certain road type, it will then be upgraded to become this road type if the current road type is of “lower” type. For example, a tertiary forest road on a cell (not very large) will be upgraded to become a secondary forest road (larger than tertiary) at a given time step if the wood flux for this road cell at the time step correspond to the “secondary” forest road type.

Once the type of a forest road is determined via the flux of wood, then the cost of the upgrade will be the cost of construction of the higher road type (base cost * multiplication value for the road type) minus the cost of construction for the current road type (see section 5.3.4). Those multiplicative costs must be relative to the type of forest road chosen to describe all of the cost parameters for the path-finding (see section 4 and further below for more details).

5.1 Simulation of road aging

The extension can simulate the aging of the forest roads as time goes by, and their potential destruction if they are not maintained. To that end, the user must input “yes” to this parameter. If so, the user must also indicate a “maximum age before destruction” value for each type of forest roads in the table of road types (see section 5.3 below).

If activated, at each time step, the forest roads will be "aged"; when their age reach a number superior to the maximum age before destruction for their category, they will be destroyed. They can be rebuilt if the algorithm identifies the pixel as the cheapest way to go to connect to a certain harvested area.

Currently, the age of the forest roads is reset when they are constructed, or upgraded via the algorithm of wood flux.

5.2 Simulation of wood flux

The extension can simulate the flux of wood going through the roads as the harvesting unfolds, and can also upgrade the type of the forest roads according to this flux. To that end, the user must input "yes" to this parameter. If so, the user must also indicate lower and upper thresholds of wood flux for every type of road in the table of road types (see section 5.3 below).

At each time step, the wood harvested in the recently harvested sites will be considered as transported toward the nearest exit sites for the wood (see below). This process is repeated for every harvested site, and the flux can thus sum on the same road cell. Once every harvested site has been considered, the algorithm will read the flux value for each forest road, and upgrade the road if the flux gets above the maximum threshold for the current road type.

5.3 Table of forest road types

This table contains every information necessary to the extension in order to take into account different forest road types.

The table should contain at least a forest road type ID, a multiplicative cost value, and a road name. If the wood flux is simulated (see previous section 5.2), it should contain thresholds for the wood flux. If the aging of forest roads is simulated (see previous section 5.1), it should contain the maximum age before road destruction.

5.3.1 Lower wood flux threshold (Optional, if wood flux is simulated)

This threshold corresponds to the minimal value of wood flux that is needed before using the corresponding type of forest road. Currently, the wood flux is expressed in number of harvested age cohorts in order to accommodate the harvest extensions. As the flux is calculated at each time step, the threshold should be expressed in age cohorts harvested by time step. The lower threshold is inclusive, meaning that any wood flux equal to a lower threshold will get included in the road type for this lower threshold.

5.3.2 Upper wood flux threshold (Optional, if wood flux is simulated)

This threshold corresponds to the maximal value of wood flux that is needed before using a higher type of forest road than the corresponding one. Currently, the wood flux is expressed in number of harvested age cohorts in order to accommodate the harvest extensions. As the flux is calculated at each time step, the threshold should be expressed in age cohorts harvested by time step. The upper threshold is not inclusive, meaning that a wood flux equal to an upper threshold will get included in the road type of the category below this threshold.

5.3.3 Forest road type ID

The forest road type ID corresponds to an integer associated with the corresponding type of forest road. These IDs should correspond to the ones that are found in the initial road network raster (see section 3.2), and will be the ones found in the output rasters (see section 2.6),

It can be any positive integer, as values inferior or equal to 0 correspond to the absence of roads. However, the forest roads types ID should not correspond to any exit point road type ID (see section 5.4), and vice versa.

5.3.4 Multiplicative cost value

This value is multiplied with the cost of construction of a forest road calculated by the algorithm to decide what is the cost of updating a forest road to the corresponding forest road type. Therefore, this multiplicative value must be relative to the cost rasters and the cost parameters given before (see sections 3 and 4).

For example, if the cost rasters and the cost parameters all concerned forest roads of type “secondary road” (that thus serves as the “reference” type of forest road), then the multiplicative cost value for this type will be 1; the multiplicative cost values for smaller forest roads (e.g. tertiary forest roads) will be inferior to 1, and superior to 1 for bigger road types (e.g. primary forest roads).

The multiplicative cost value must be superior to 0.

5.3.5 Maximum road age before destruction (Optional, if road aging is simulated)

This value corresponds to the number of years that the corresponding forest road type is supposed to last before being destroyed by senescence. It must be superior to 0.

5.3.6 Road name

A string of characters without any spaces corresponding to the name of the corresponding forest road type.

5.4 Table of exit points for the wood

This table contains every information necessary to the extension in order to take into account different exit points for the wood (e.g. sawmills, main paved road network, warehouse, etc.).

The table must only contain IDs and names associated to exit points for the road. These are crucial, as the algorithm will need to connect to them in order to successfully reach a harvested area. At least one of these exit points must be present in the initial road raster (see section 3.2). The IDs for the exit points must be different from IDs used forest roads, and also from -1 and 0 which are reserved by the algorithm.

6 Example parameter file

```
>> To be read properly, the parameter file must contain the parameters in this order.

>>------------------------------------------------------------------------
>> BASIC PARAMETERS

LandisData "Forest Roads Simulation"

Timestep 10

HeuristicForNetworkConstruction Closestfirst

SkiddingDistance 250

LoopingBehavior Yes
LoopingDistance 750
LoopingMaxPercentageOfRoads 5
LoopingMaxCost 10

OutputsOfRoadNetworkMaps ./output/disturbances/roads/roadNetwork.tif
OutputsOfRoadLog ./output/disturbances/roads/


>>------------------------------------------------------------------------
>> INPUT RASTERS AND COST PARAMETERS
>> Only the initial road network raster and the distance cost are
>> essential. If you do not want to use one of the cost for the path-
>> -finding, just indicate "none" as the parameter value for the raster
>> location, and "0" for the value of the associated cost.

RasterOfBuildableZones "../../sharedRasters/buildable_zones.tif"
InitialRoadNetworkMap "../../sharedRasters/initial_road_network_empty.tif"
DistanceCost 365.6

CoarseElevationRaster "../../sharedRasters/coarse_elevation.tif"

CoarseElevationCosts
>> Lower elevation      Upper elevation        Additional
>>    threshold            threshold             value
         0                    9                    0
         9                    16                   1930
         16                   41                   3454
         41                   1000                  100000

FineElevationRaster "../../sharedRasters/fine_elevation.tif"

FineElevationCosts
>> Lower elevation      Upper elevation      Multiplication
>>    threshold            threshold             value
        0                     300               1
        300                   600               1.5
        600                   1000              2
        1000                  10000             10

CoarseWaterRaster "../../sharedRasters/coarse_hydrology.tif"
CoarseWaterCost 1400000

FineWaterRaster "../../sharedRasters/fine_hydrology.tif"
FineWaterCost 20000

SoilsRaster "../../sharedRasters/soils_cost.tif"


>>------------------------------------------------------------------------
>> ROAD TYPE THRESHOLDS AND MULTIPLICATION VALUES
>> These parameters are all essential to the functioning of the
>> extension.
```

```
SimulationOfRoadAging Yes
SimulationOfWoodFlux  Yes

RoadTypes
>> Lower Wood Flux  Upper Wood Flux    Road type   Multiplicative       Maximum age         Road Type
>>    threshold        threshold          ID        Cost Value      Before destruction      Name
  0                   100                6            0.31                 1                Winter_roads
  100                 500                5            0.46                 3                Class_05
  500                 1000               4            0.66                 10               Class_04
  1000                5000               3            1                    15               Class_03
  5000                10000              2            2.88                 25               Class_02
  10000               1000000            1            4.26                 25               Class_01
  1000000             1000000000         7            4.73                 50               Out_of_norm

RoadTypesForExitingWood
>> Road type   Road Type
>>    ID        Name
      8         Sawmill
      9         MainRoadNetworkPaved
```