Forest Roads Simulation v0.5
LANDIS-II Extension
User Guide

Clément Hardy[1]

[1]Université du Québec à Montréal

Last Revised: August 20, 2019

Table of Contents

1 Introduction

This document describes the Forest Roads Simulation ('plug-in') for the LANDIS-II model. Users should read the *LANDIS-II Model User's Guide* prior to reading this document.

1.1 The road Landscape

The Forest Roads Simulation extension 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 Forest Roads Simulation extension works in tandem with a harvest extension.

To function, the Forest Roads Simulation extension edit 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 read when the simulation is completed.

1.1.1 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, but they do not contain a road.

1.1.2 Sites with sawmills or with pieces of the main 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 extension as places where the wood harvested in the landscape can "exit" the landscape. Every initial road landscape must have at least one site of this category; if not, the algorithm that determine the type of the different forest roads (see below) will not work.

1.1.3 Sites with forest roads

Finally, the sites can be indicated as containing a forest road used to transport timber from the harvested sites to sawmills or to the main road network. These forest roads can be of different type: Primary, secondary, tertiary, winter or temporary. The type of the forest road can be indicated by the user in the raster input file for the road network; or, it will be calculated as the network evolves with an algorithm that treats the flux of timber as a flux of water (see below).

1.2 Determining the type of the forest roads

Forest roads can be described as different categories, relative to their dimensions, their materials, and so on. In reality, forest roads of higher categories are only constructed to facilitate the transport of timber as they are costly to build. 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 hydrologic 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, 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 from which a certain forest type must be built to allow for transportation.

1.3 Calculating the least-cost path and the cost of construction

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 Targest 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 softwares for designing forest road networks such as REMSOFT (Remsoft, 2019) use a heuristic coupled with a path-finding algorithm to solve the MTAP as a sequence of STAP ordered via the heuristic. For example, the heuristic can be “deal with the farthest harvest site from current roads first”; and the STAP algorithm can be a path-finding algorithm 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, in order of their distance to existing roads. The heuristic chosen will thus heavily influence the shape of the 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. 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 elements that are important in the creation of a forest road: distance, slope (elevation), existing bodies of water (rivers, lakes), soils and vegetation. While it would be impossible to propose an operational path for forest roads with the discrete resolution used by LANDIS-II, a global idea of the cost of a 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 parameters described previously); and the type of the road determined by the “timber flux” algorithm (see previous sub-section).

1.4 Major Releases

No major release has been made yet; the extension is still in construction, and many of the main functions are currently missing.

1.5 Minor Releases

1.5.1 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 timestep, the extension gets all of the recently harvest sites, and construct a 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 timestep to see the evolution of the road network in the landscape

1.6 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/

1.7 Acknowledgements

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 The input rasters

To function, the Forest Roads Simulation extension 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 have to be in a certain format to be read and understood by the extension, however. In this section, you will find the details and some indication of how to generate those rasters quickly.

The Forest Roads Simulation extension cannot work without an initial road network raster containing at least one pixel/site of the category “Sawmill” or “Main road network” (see below). Every other input raster is optional. Yet, without them, the path of the roads will 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 throw a warning to the user, and just not use the associated variable 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. Remember that by defaults, roads can be constructed on any site in this extension (active, or not active).

2.1 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. Six values are recognized as road types, while all of the other correspond to an empty site:

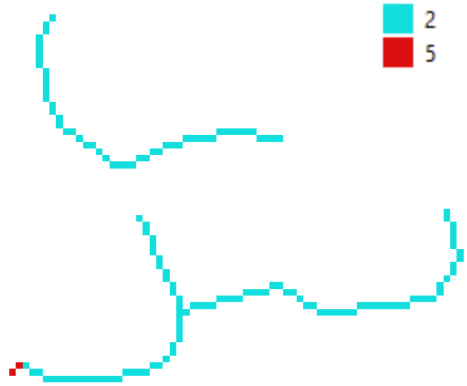
Raster pixel value	Road type
1	Primary road
2	Secondary road
3	Tertiary road
4	Winter/temporary road
5	Sawmill
6	Main road network (paved)
7	Undetermined (but there is a road)
Any other value	No roads or unrecognized

The same values are used in the output rasters generated by the extension.

To generate such a 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 (1, 2, 3, 4, 5, 6, 7). 0 should be used as the “no road” value.

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 raster only contains two pixels/sites indicated as a sawmill (5), and pixels/sites indicated as secondary roads (2). The rest will thus be sites without roads.

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.

2.2 The coarse elevation raster

One of the main parameters to take into account during the construction of roads is the slope of the land; it is always more complex to build a road on a high slope than on a flat terrain.

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 information. The first is the “coarse” elevation raster.

In this raster, each pixel 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.

2.3 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 road, requiring detours or an increase of road distance to traverse the pixel/site.

To create such a raster, the user can 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 wanted raster. Secondly, the “Sum line length” tool to fill an attribute field of the grid with the length of the contours line 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 2.1). Every value can be used, but should not reach above the int16 limit (32,768). We thus recommend that a measure in meters should be used.

2.4 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 will be 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 burn the following values in the pixel of the raster.

Raster pixel value	Body of water

0	No body of water
1	River
2	Lake

2.5 The fine water raster

This second water raster will contain information about streams. We recommend to make it the same way as the fine elevation raster, but by using polylines that describe the position of streams.

Thus, the corresponding raster will 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 doesn't 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, upping the costs of constructing the forest road in that pixel/site.

2.6 The soil raster

This last raster will give information to the algorithm concerning the soil of the landscape. This can be important for the construction of roads, especially if the soils allow the builders and engineers to use local material (gravel, etc.) to build the 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 should be a mapcode; it can be the mapcode of an ecoregion from the ecoregions raster (in that case, the user can indicate the ecoregion raster as the soil raster).

For each mapcode present in the raster, the user will have to indicate in a data table if this type of soil allows the use of local materials to construct the road (yes/no), and if not, a supplementary cost to build a road in the given pixel/site.

3 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 said raster 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 and values that are easy to discuss or relate too, making the parametrization and the interpretation of results easier with both modelers and non-modelers.

We also recommend all of these costs to be expressed for one chosen type of forest road category, like a primary, secondary or tertiary road. Then, a multiplication parameter will be applied when roads of lower or higher will be constructed to augment or reduce the cost.

3.1 Distance cost

This parameter must follow the location of the initial network raster, and is the only obligatory one to be precise. This parameter will represent the unreducible cost of building a 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 unreducible cost of building 100m of forest road.

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 road; taking topography into account; etc.) in order for the cost outputs to be more realistic.

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

3.2 Coarse elevation cost

This cost corresponds to an addition to the basic distance cost due to the necessity of constructing a road upward or downward. It is expressed as a single value that will be multiplied with the difference between the elevation of one site and its neighbor, and then added to the distance cost of transition between the two sites.

The parameter should thus be a single value superior to 0. If no coarse elevation raster have been inputed (the parameter has been set as “none”), then this cost will not be taken into account during the path-finding process.

3.3 Fine elevation cost

This cost corresponds to a multiplication of the basic distance cost due to the fact that the road will have to slalom in order to avoid fine topographic elements (cliffs, depressions, etc.), thus extending the distance of 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 basic distance cost according to fine topography value of the sites to cross.

The thresholds should thus be values 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 value a single value superior to 0.

3.4 Coarse water cost

This cost will simply be the cost of building a bridge in 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 of 2 000 000$, and if the side of a site in the LANDIS-II landscape is of 50m, then this cost should be of 1 000 000$.

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

3.5 Fine water cost

This cost is the cost associated with the crossing of a stream, which will necessitate the construction of a culvert. The user just has to input the mean cost of a culvert. 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 will have to be taken into account by the cost algorithm.

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

3.6 Soil cost

Finally, the soil cost will be an additive cost corresponding to the soil mapcode contained in the soil raster. This parameter will take the form of a table of value; for each mapcode, a value must be given, that will be added to the cost of crossing a site with the given mapcode.

The values of this cost for each corresponding map code should be superior to 0.

4 Road type thresholds and cost multiplication values

As indicated in sub-section 1.2, the type of the forest roads to construct is deduced by the use of an algorithm that considerers the forest road network as a kind of watershed. In this view, each recently harvested site becomes a source of wood at the considered timestep. To simplify things and to not burden the user with a complex parametrization, each harvested site is considered to provide one arbitrary unit of timber.

Once all of the roads are determined at the current timestep to access the recently harvested site, the algorithm will then determine:

- What type should be the newly constructed roads at this timestep
- If other roads need to be upgraded to accommodate the flux of timber at this timestep in order to carry it to the closest “exit point” (sawmill, or main road network).

To do this, the algorithm will make the timber from the harvested site “flow” from road to road, until it reaches an exit point. The type of the roads to be constructed will be determined by the road type thresholds. For a given road, if the additive flux of timber that goes through this road is between the thresholds of a certain road type, it will then become this road type.

Once the type of a road is determined via the flux of timber, then the base cost of construction will be multiplied by the cost multiplication value associated with the given type of road. This multiplication value must be relative to the type of road chosen to describe all of the cost parameters for the path finding (see section 3).

An existing road can also be upgraded to a higher road type. In such a case, the cost of the upgrade will be the cost of construction of the higher road type (base cost x multiplication value for the road type) minus the cost of construction for the current road type.

4.1 Primary road threshold

This parameter is the value of timber flux above which the concerned road will have to become a primary road to accommodate for the flux. Every flux of timber above this value will automatically result in the concerned roads becoming a primary road. Primary forest roads are the largest, most expensive type of forest roads; but they are also the ones that can accommodate the most timber flux.

4.2 Primary road cost multiplication value

This parameter’s value will multiply the base cost of construction of a given road when it needs to become a primary road. It will also be applied in case of upgrading from a road of lower type (e.g. secondary or primary road).

4.3 Secondary road threshold

This parameter’s is the value of timber flux above which the concerned road will have to become a secondary road to accommodate for the flux, as long as it is below the primary road threshold. Every flux under this value will result in a tertiary or temporary road type (see below).

4.4 Secondary road multiplication value

This parameter is similar to the primary road cost multiplication value (see sub-section 4.2), but should reflect the cost of a secondary road.

4.5 Temporary road percentage

This parameter will indicate to the model which percentage of the tertiary roads can be temporary roads or winter roads in the landscapes. Temporary roads only last for the current timestep, and will not be usable at the next timestep to reach harvested sites. However, they should be less costly to build than permanent tertiary roads.

4.6 Tertiary road multiplication value

This parameter is similar to the primary road cost multiplication value (see sub-section 4.2), but should reflect the cost of a tertiary road.

4.7 Temporary road multiplication value

This parameter is similar to the primary road cost multiplication value (see sub-section 4.2), but should reflect the cost of a temporary road.

5 Other input parameters

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

5.1 LandisData

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

5.2 Timestep

This parameter is the extension's timestep. Value: integer > 0. Units: years.

5.3 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 1.3). 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 softwares, is used by the extension instead.

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

We recommend the user to use different scenarios with different heuristic to determine if this affects their simulated landscape or not.

5.4 Skidding distance

The skidding distance indicates the distance above which it becomes necessary to construct a road to access a site that is harvested. If the harvest site is in such a range to the closet site with a road on it, then no road will be constructed to this harvested site. It will be supposed that the wood will be transported by skidding.

5.5 Location of the outputs rasters

This parameter indicates where the user wants the ouputs rasters generated at each timestep to be located. It can be expressed as a relative or an absolute path.

The name of the ouput rasters must be indicated, with the “.tif” extension. The Forest Roads Simulation extension will then automatically add the number of the timestep at which the output raster was made at the end of the raster’s name.

5.6 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.

6 Example parameter file

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

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

LandisData "Firest Roads Simulation"

Timestep 15

HeuristicForNetworkConstruction Closestfirst

SkiddingDistance 150

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.

InitialRoadNetworkMap ./input/disturbances/roads.tif
DistanceCost 10000

CoarseElevationRaster ./input/disturbances/coarse_elevation.tif
CoarseElevationCost 100

FineElevationRaster ./input/disturbances/fine_elevation.tif
FineElevationCosts
>> Lower elevation      Upper elevation      Multiplication
>>    threshold            threshold              value
          0                   400                  1
         400                  2000                 1.5
         2000               10000000               2

CoarseWaterRaster ./input/disturbances/coarse_water.tif
CoarseWaterCost 1000000

FineWaterRaster ./input/disturbances/fine_water.tif
FineWaterCost 10000

SoilsRaster ./input/disturbances/soils.tif
SoilsCost
>> Map Code       Additional Cost
```

```
      1                 0
      2                 1000
      3                 10000
      4                 2000

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

PrimaryRoadThreshold 10
PrimaryRoadMultiplication 5

SecondaryRoadThreshold 5
SecondaryRoadMultiplication 2

TemporaryRoadPercentage 40
TertiaryRoadMultiplication 1
TemporaryRoadMultiplication 0.6
```