**Instructions for LUSOvar, including new LUSO features**

**Michael Renton**

**Plant Biology, University of Western Australia**

These are some basic instructions for LUSOvar, the stochastic or ‘variability’ version of LUSO.

Please note that LUSOvar is still very much under development – if you have problems or for the most recent information, please contact Michael Renton or Roger Lawes

Before you start, you should ensure you are reasonably familiar with the original (non-stochastic) functionality of LUSO. If you are not, first look at the LUSO tutorial and pdf articles in the docs folder. You also need to make sure python and required libraries are installed on your computer. See the ‘install python.txt’ file for more instructions if needed.

LUSOvar allows new analyses that account for variable seasons, instead of assuming an average season every year, as does the original LUSO. It allows the user to 1) simulate a single given sequence of land uses over a given sequence of season types, 2) find the optimal sequence of land uses for a given sequence of season types, 3) simulate a single given sequence of land uses over a large number of randomly selected sequences of season types, to predict the range of possible outcomes for that land use sequence.

**Python files for performing LUSO and LUSOvar analyses**

This is a list and summary of the python files that provide ‘front-end’ functionality for performing LUSO and LUSOvar analyses. They can run by simply double-clicking the icon, with output data being saved in the relevant file in the outputs directory. Alternatively, for more control and information, they can be opened in an editor such as IDLE and run from there. In all these files relevant information is specified in the first few lines of code (eg season sequence to simulate, landuse sequence to simulate etc), so this is the part that the user is most likely to want/need to edit. All these files are found in the base LUSO directory; other files in the ‘files’ directory are less likely to be of direct use or interest to most users, although some of them provide required ‘back-end’ functionality for these ‘front-end’ files to work.

**single run.py** should work as in the orginal LUSO, simulating a specified land use sequence assuming an average season every year, and give the biological and economic results.

**single run variable season.py** should simulate a specified land use sequence for a specified sequence of seasons, and give the biological and economic results. ‘Pure random’ variation (ie beyond seasonal effects) can be turned off if required, to ensure that this simulation gives the same results each time it is run.

**luso.py** should work as in the orginal LUSO, finding the best land use assuming an average season every year, and give the biological and economic results of this best result.

**luso variable season.py** should find the best land use for a specified sequence of seasons. If the specified sequence were all average seasons then it should work the same as luso.py. ‘Pure random’ variation (ie beyond seasonal effects) is turned off by default, to enable optimisation to work.

**multiple runs variable season.py** simulates a given crop rotation for a specified number of times, with each time having a randomly chosen sequence of seasons. The seasons within the sequence can be chosen with or without replacement (as specified at the top of the file)

**Required csv parameter files (see inputs directory)**

Modification of the following files is the main way that the user changes the inputs to the LUSO model and analysis tools. In some cases they may also need to change parts in the top of the python files (see above).

'\_LUSdetails\_used.csv': all land-use-specific information for an average season

'\_parameters\_used.csv': all non-land-use-specific information for an average season

'\_stochasticParameters\_used.csv': all information on how parameters vary in other-than-average seasons – this is NOT needed to run the basic LUSO analyses, but is needed to run LUSOvar analyses.

**Optional csv parameter files (see inputs directory)**

'\_additional\_effects.csv': this file allows the specification of additional effects of a landuse on the yield of any subsequent landuse. It is flexible, because the initial land use, the effected landuse, the number of years between them, and the multiplicative effect can all be specified for as many combinations as desired. This means that landuses can affect different types of landuse in different ways, and that the effect can be different at different ‘lags’. This contrasts with the simpler ‘water multiplier’ option where a landuse can effect just the following landuse (1 year separation), and all types of landuse are affected in the same way. For example, if the file looked like the example below, then wheat following a fallow would have a 20% yield boost, canola following a fallow would have a 30% yield boost, and canola two years after a fallow would have a 10% yield boost. If this file is not present, or contains no lines beyond its header, then no additional effects are applied.

|  |  |  |  |
| --- | --- | --- | --- |
| initiallu | laterlu | yearsbetween | effectonlaterlu |
| fallow | wheat | 1 | 1.2 |
| fallow | canola | 1 | 1.3 |
| fallow | canola | 2 | 1.1 |

'\_disallowed\_combinations.csv': this file allows the specification of disallowed or penalised combinations of landuses. For example, if the file looked like the example below, then hi wheat following four other high wheats would be penalised by $99, while canola following canola would be penalised $9999. Specifying a large penalty effectively disallows the combination, whereas a small penalty may be used to specify a realistic additional cost, such as additional pest or disease control. If this file is not present, or contains no lines beyond its header, then no penalties are applied.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| penalty | combination | |  |  |  |
| 99 | hi wheat | hi wheat | hi wheat | hi wheat | hi wheat |
| 9999 | canola | canola |  |  |  |

‘\_price\_variability.csv’: this file allows the specification of price variability, in addition to seasonal variability effects on biological factors. For example, if the file looked like the example below, then in any given season wheat price (and hi wheat price) would have equal chance of being multiplied by 1.1, 1 or 0.9 times its default value species in '\_LUSdetails\_used.csv', while canola would have an equal chance of being multiplied by any of the five numbers given. If this file is not present, or contains no lines beyond its header, then no price variability is applied. For any landuse not given in this file, price will always be the default value. This file is only used (and only useful) with the multiple runs variable season.py analysis (see above).

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| landuse | multipliers | |  |  |  |
| wheat | 1.1 | 1 | 0.9 |  |  |
| canola | 1.2 | 1.1 | 1 | 0.9 | 0.8 |
| hi wheat | 1.1 | 1 | 0.9 |  |  |

**Other new additions to LUSOvar May 2013**

In addition to major new additions described above, there are a number of other new features in LUSOvar added in May 2013:

* running the **multiple runs variable season.py** analysis produces a histogram of profits in a figure in the outputs directory (it also produces a csv file with all the profits for plotting in other way as the user wishes)
* there are now extra columns in the ‘details’ output files, including one for undiscounted profit for each year, to increase transparency on how profit is calculated
* profit calculation function has an option to calculate ‘annualised’ profit (average annual profit weighted by discounting) as well as total profit over the full sequence – this facilities comparison between sequences of different length
* water effects – as well as the flexible ‘additional effects’ optional file described above, there is a new column in the '\_LUSdetails\_used.csv' file that allows the user to specify that a landuse has a certain effect on the yield of any following landuse/crop (through giving more stored soil moisture for example)

**How is disease modelled in LUSOvar?**

There are two variables tracked from year to year: disease incidence DI and disease damage DD.

For flexibility, the functions that control how DI changes from year to year, and the way that DD depends on DI and other factors are specified in their own file: diseasefuncs.py

In the current disease model, both variables are constrained to always lie between zero and one by using a logistic scale.

**Disease incidence DI**

Disease incidence DI is a representation of the amount of the disease-causing organism early in the growing season. It is always between 0 and 1, but could be scaled to any reasonable scale as desired. DI can thought of as a quantile ie 0.5 is an average incidence (whatever that may be), and close to 1 is the maximum incidence.

Incidence DI can be affected by

* the incidence from the previous year (DI\_prev)
* the previous crop (NOT by the current crop, since it is early season incidence)
* the season (probably the summer weather mostly, or the early growing season weather from this year and maybe the late growing season weather from last year)
* a random factor – this represents anything for which we have no other explanation

The equation used is:

logit(DI) = IE\_previnc \* logit(DI\_prev) + log(IE\_prevcrop) + log(IE\_season) + rnorm(0,IE\_random)

IE is abbreviation for ‘incidence effect’

IE\_previnc is a general LUSO parameter that should be between zero and one. If set to zero then previous incidence has no effect at all on new incidence. If set to one then previous incidence has full effect on new incidence. If between zero and one, then previous incidence has partial effect on new incidence, the closer to zero, the less effect. The setting of this parameter affects whether other parameters control direction of movement of disease or absolute level of disease.

IE\_prevcrop is a land-use specific parameter that controls the effect of the previous landuse. It must be greater than zero. If equal to one then incidence tends to be average after this landuse/crop, or to maintain its previous level. If less than one then incidence tends to be low after this landuse/crop, or to reduce its previous level. If greater than one then incidence tends to be high after this landuse/crop, or to increase its previous level.

IE\_season is a season specific parameter that gives the expected incidence, given the previous weather (seasonal effect), all else being equal. It must be greater than zero. If equal to one then incidence tends to be average after this season type, or to maintain its previous level. If less than one then incidence tends to be low after this season type, or to reduce its previous level. If greater than one then incidence tends to be high after this season type, or to increase its previous level.

The general parameter IE\_random controls the size of the random component in DI. If IE\_random is set to zero then there is no random effect at all, if IE\_random is set to be small (say 0.05) then there is a small random effect and if IE\_random is set to be large (say 0.5) then there is a large random effect.

As mentioned above, DI is constrained to always lie between zero and one by using a logistic scale. In addition, DI is constrained to be at least DI\_min, where DI\_min is a general LUSO parameter, to represent the fact that there is always some background population of the pathogen.

**Disease damage DD**

Disease damage DD is the proportion of crop yield lost to disease, for example, if DD = 0.02 then yield is reduced by 2% because of disease.

As mentioned above, DD is constrained to always lie between zero and one by using a logistic scale.

Incidence DD can be affected by

* the current incidence
* the current crop
* the season (probably the growing season weather mostly, especially the finish)
* a random factor – this represents anything for which we have no other explanation

The equation used is:

logit(DD) = DE\_inc \* logit(DI) + logit(DE\_crop) + log(DE\_season) + rnorm(0,DE\_random)

DE is abbreviation for ‘damage effect’

DE\_inc is a general LUSO parameter that should be zero or greater. If set to zero then incidence has no effect at all on damage. If set to one then incidence has an approximately linear effect on damage.

DE\_crop is a land-use specific parameter that represents the ‘average’ damage to this kind of crop, given an average season and an average incidence. It should be between one and zero. If set to zero, no damage will ever occur, if set to one then damage is always 100%.

DE\_season is a season specific parameter that represents the seasonal effect on damage. It should be greater than zero. Setting it to 1 means season has no effect, ie an average season. Values close to zero mean damage will be increased in this kind of season and setting it high means damage will be increased in this kind of season (eg. for low incidence levels, a value of two means approximately doubling damage).

The general parameter DE\_random controls the size of the random component in DE. If DE\_random is set to zero then there is no random effect at all, if DE\_random is set to be small (say 0.05) then there is a small random effect and if DE\_random is set to be large (say 0.5) then there is a large random effect.