

Decision Analysis of agroforestry options reveals adoption risks for resource-poor farmers

Implementation of the decision model for agroforestry planning in Northwest Vietnam

Hoa Do^{1*}, Eike Luedeling¹, Cory Whitney¹

2020-03-26

¹Department of Horticultural Sciences, Institute of Crop Science and Resource Conservation, University of Bonn, Germany

***Corresponding author:** *dohoa1190@gmail.com*

Journal: *Agronomy for Sustainable Development*

This document illustrates the development of the mathematical model used to calculate Net Present Values (NPV) of agroforestry interventions promoted in Northwest Vietnam. This process was implemented in the R programming language (R Core Team 2019). To illustrate the modeling procedures we highlight NPV calculations for the decision to choose coffee-based agroforestry systems (combinations of coffee, teak, plum, annual crops and grass) over conventional monocultures of maize. The same procedures and functions can be used for the other interventions. The corresponding input data tables for each system are provided in the rest of the supplementary materials.

Model development process

We employed the `decisionSupport` R package (Luedeling, Goehring, and Schiffrers 2019) and its functions to develop the model and to perform probabilistic simulations.

The `chance_event()` function was used to calculate the time series occurrence of risk events. This function requires two main arguments:

1. the probability that a particular risk event will happen in a given year and
2. the time line for the simulation.

The `gompertz_yield()` function was used to calculate the yields of tree crops using identified parameters given an assumed pattern.

The value varier function `vv()` was used to produce realistic time series of data that included year-to-year variation around a specified mean, determined by a desired coefficient of variation. Both mean and coefficient of variation are provided to the model via the input table.

The `discount()` function was used to calculate NPV in the last step of the decision modeling procedure. The function implements discounting of future costs or benefits by a decision maker-specific discount rate, which is also specified in the input tables.

The `decisionSupport()` function was used to conduct the full decision analyses by performing probabilistic simulations using the Monte Carlo technique. Monte Carlo simulation can perform calculations using distributions and ranges. This is a major advantage over traditional spreadsheet calculations. In a Monte Carlo simulation, a particular calculation is repeated many times, with each run parameterized with random draws from distributions that are defined for each variable. The output then consists of a set of values instead of single estimates.

The function requires two inputs:

1. a data table (.csv format) specifying the names and distributions for all uncertain variables and,
2. an R function that predicts decision outcomes based on the variables named in the data table.

We defined binomial distributions for the occurrence of risk events, since the respective events could only either occur or not occur. Most of the yield-related variables, discount rates and the benefits from soil loss prevention were expressed as normal distributions but constrained to positive values only. Certain highly unpredictable variables such as crop prices were expressed using uniform distributions. Other highly consistent variables such as tree planting densities were assigned constant values.

The `multi_EVPI()` function was used to calculate the expected value of perfect information (EVPI) for multiple variables in the model using the simulation outcome table from the `decisionSupport()` function.

First we defined sample values for all variables by generating a `make_variables` function to select randomly from each variable in our input table.

```
make_variables <- function(est,n=1)
{x <- random(rho=est,n=n)
for(i in colnames(x))assign(i, as.numeric(x[1,i]),envir=.GlobalEnv)}
```

We applied the `make_variables` function to `estimate_read_csv`, the input data table, to generate one random number for each variable. These numbers are not used in the simulation, but they allow running all the code in the model for testing and troubleshooting purposes.

```
make_variables(estimate_read_csv(paste("Input tables/CAFS_teak.csv",sep="")))
```

The mathematical model could then be coded. We gave the name `AF_benefit` to the function that calculates the NPV.

```
AF_benefit <- function(x,varnames)
{
#Define each variable as vectors of 25 values corresponding to 25 years of simulation

mono_costs <- rep(0,n_years) #the annual cost of maize monoculture
establishment_cost <- rep(0,n_years) #costs for setting up agroforestry systems (AF)
maintenance_cost <- rep(0,n_years) #the annual cost spent on AF
timber_yield <- rep(0,n_years) #the yield of teak
tree_yield <- rep(0,n_years) #the yield of plum
coffee_yield <- rep(0,n_years) #the yield of coffee
AF_maize <- rep(0,n_years) #the yield of maize in AF
mono_maize <- rep(0,n_years) #the yield of maize in monocultures
grass_benefit <- rep(0,n_years) #the benefit from grass
annual_benefit <- rep(0,n_years) #the benefit of annual crops
reduced_soil_loss <- rep(0,n_years) #amount of soil loss reduced in AF
#compared to monoculture systems

#Simulate the chance for different events to occur during the simulation period
frost <- chance_event(coffee_risk,value_if = 1,n=n_years)
tree.risk <- chance_event(tree_risk,value_if = 1,n=n_years)
maize.risk <- chance_event(maize_risk,value_if = 1,n=n_years)

#Number of timber harvests throughout 25 years of simulation
timb_harvest <- as.numeric(1:n_years%round(freq_tim_harvest)==0)

#Calculate system costs and convert currency from VND to USD using currency
#exchange rate (cur_change) applied for any variable with monetary value
 #(i.e. costs and benefits)
maintenance_cost[1] <- 0
```

```

maintenance_cost[2:4] <- vv(main_cost_stage1,CV_cost,3)/cur_change
maintenance_cost[5:10] <- vv(main_cost_stage2,CV_cost,6)/cur_change
maintenance_cost[11:25] <- vv(main_cost_stage3,CV_cost,15)/cur_change

establishment_cost[1] <- establis_cost/cur_change
establishment_cost[2:25] <- 0

#Calculate system benefits

#benefit of timber
timber_benefit <- (timb_benefit*timb_harvest)/cur_change

#benefit of trees
tree_yield <- gompertz_yield(max_harvest = max_tree_harvest,
                             time_to_first_yield_estimate = time_first_tree_est,
                             time_to_second_yield_estimate = time_sec_tree_est,
                             first_yield_estimate_percent = first_tree_est_per,
                             second_yield_estimate_percent = sec_tree_est_per,
                             n_years=n_years,
                             var_CV=CV_tree_yield,
                             no_yield_before_first_estimate = FALSE)
tree_yield <- tree_yield*(1-tree.risk*yield_tree_risk)
tot_tree_yield <- tree_yield*num_of_tree
tree_benefit <- (tot_tree_yield*tree_price)/cur_change

#benefit of coffee
coffee_yield[1:15] <- gompertz_yield(max_harvest = max_coff_harvest,
                                     time_to_first_yield_estimate = time_first_coff_est,
                                     time_to_second_yield_estimate = time_sec_coff_est,
                                     first_yield_estimate_percent = first_coff_est_per,
                                     second_yield_estimate_percent = sec_coff_est_per,
                                     n_years=15,
                                     var_CV = CV_coff_yield,
                                     no_yield_before_first_estimate = FALSE)

coffee_yield[16:25] <- coffee_yield[6:15]

#account risk of frost in the calculation of coffee yield
coffee_yield <- coffee_yield*(1-frost*coff_frost_yield)

tot_coffee_yield <- coffee_yield*num_coff_tree
coffee_benefit <- (tot_coffee_yield*vv(coff_price,CV_coff_price,n_years))/cur_change

#benefit of maize in AF systems
time <- 1:n_years
decay_speed_AF <- -log(1-decay_rate_AF)
AF_maize <- min_AFmaize+(max_AFmaize-min_AFmaize)*exp(-decay_speed_AF*(time-1))
tot_AF_maize <- vv(AF_maize*(1-maize.risk*yield_maize_risk),CV_maize_yield,n_years)
AF_maize_benefit <- (tot_AF_maize*vv(maize_price,CV_maize_price,n_years))/cur_change

```

```

#benefit of maize in monoculture
decay_speed_mono <- -log(1-decay_rate_mono)
mono_maize <- max_monomaize*exp(-decay_speed_mono*(time-1))
tot_mono_maize <- vv(mono_maize*(1-maize.risk*yield_maize_risk),CV_maize_yield,n_years)
mono_revenue <- (tot_mono_maize*vv(maize_price,CV_maize_price,n_years))/cur_change
mono_costs <- vv(mono_cost,CV_cost,n_years)/cur_change
mono_benefit <- mono_revenue-mono_costs

#benefit of grass
grass_benefit <- vv(grass_profit,CV_grass_profit,25)/cur_change

#benefit of soybean
annual_benefit[1:3] <- annual_benefit/cur_change
annual_benefit[4:25] <- 0

#benefit of soil erosion control
reduced_soil_loss[1] <- 0
reduced_soil_loss[2:10] <- reduced_percents1*mono_soil_loss
reduced_soil_loss[10:25] <- reduced_percents2*mono_soil_loss
soil_saved <- vv(reduced_soil_loss,CV_eros_control,n=n_years)
erosion_benefit <- (soil_saved*soil_saved_payment)/cur_change

#Calculate NPVs of AF and monoculture and NPV of AF vs. monocultures of maize
system_benefit <- timber_benefit+tree_benefit+coffee_benefit+AF_maize_benefit+
  grass_benefit+annual_benefit+erosion_benefit #total revenue of AF

bottomline_benefit <- system_benefit-maintenance_cost-establisment_cost

tradeoff_benefit <- bottomline_benefit-mono_benefit

cash_flow <- discount(bottomline_benefit,discount_rate=discount_rate,
  calculate_NPV = FALSE)

cum_cash_flow <- cumsum(cash_flow)
NPV_mono <- discount(mono_benefit,discount_rate=discount_rate,
  calculate_NPV = TRUE ) #NPV of monoculture system
NPV_tradeoff <- discount(tradeoff_benefit,discount_rate=discount_rate,
  calculate_NPV = TRUE )
#NVP of the decision to choose AF or monocultures
NPV_system <- discount(bottomline_benefit,discount_rate=discount_rate,
  calculate_NPV = TRUE)#NVP of AF system
return(list(cashflow=cum_cash_flow,trade_off=NPV_tradeoff,
  NPV_maize=NPV_mono,NPV=NPV_system))
}

```

The probabilistic simulation was implemented using the `decisionSupport` function to return a Monte Carlo data table with 10,000 values for each of the input and response variables in the return list. A VIP score is generated for each input variable of the PLS regression.

```

decisionSupport(inputFilePath = paste("Input tables/CAFS_teak.csv",sep=""),
  outputPath = paste("MCResults/CAFS_teak",sep=""),
  write_table = TRUE,

```

```
welfareFunction=AF_benefit,
numberOfModelRuns=10000,
functionSyntax="plainNames")
```

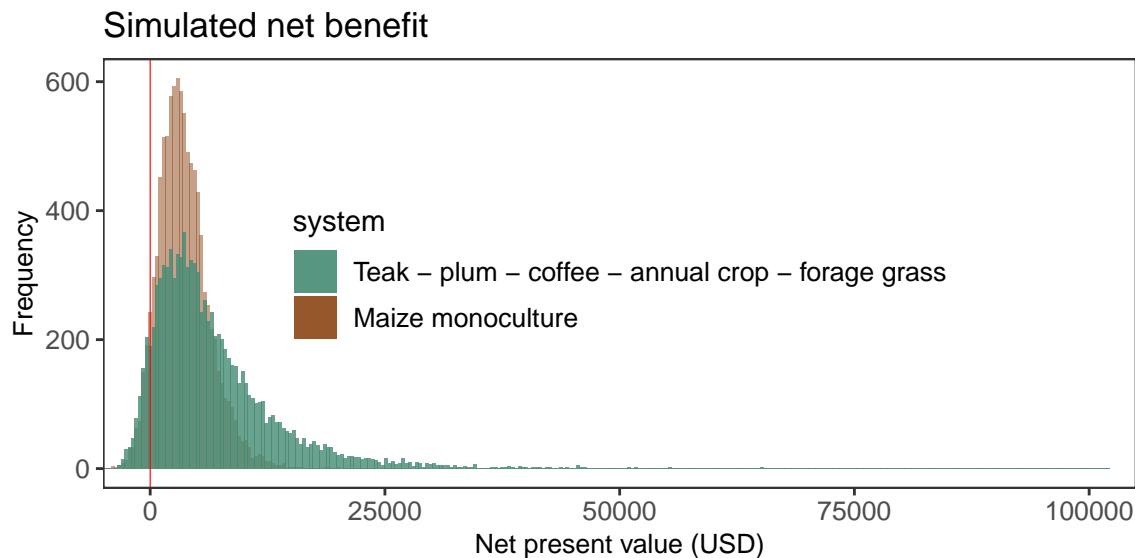
The outcome table is then used to calculate the Value of Information for uncertain variables using the `multi_EVPI` function in the `decisionSupport` package (Luedeling, Goehring, and Schiffrs 2019):

```
MCall <- read.table(file="MCResults/CAFS_teak/mcSimulationResults.csv",
                    header = TRUE, sep=",")
mc <- MCall[,c(2:50,77)] # extracting dataset considering only initial input variables
                        #and the NPV of the decision of choosing AF or monocultures

multi_EVPI(mc, "trade_off", write_table=TRUE)
```

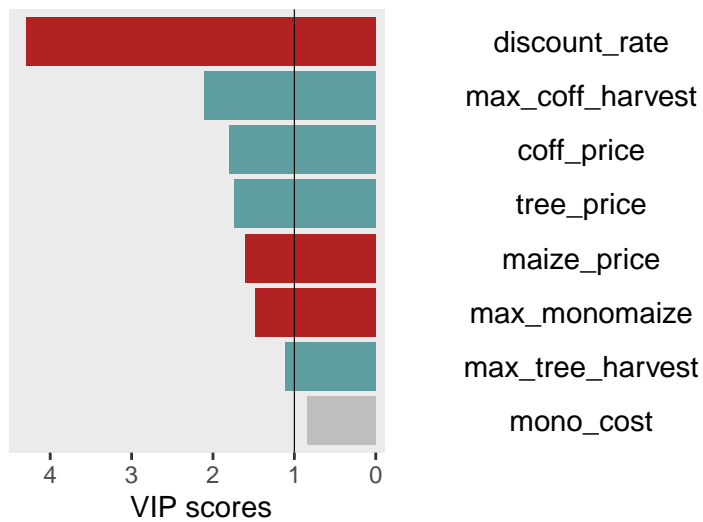
Data visualization

The simulated NPVs of agroforestry and maize monoculture are visualized using the `ggplot2` package (Wickham et al. 2020)

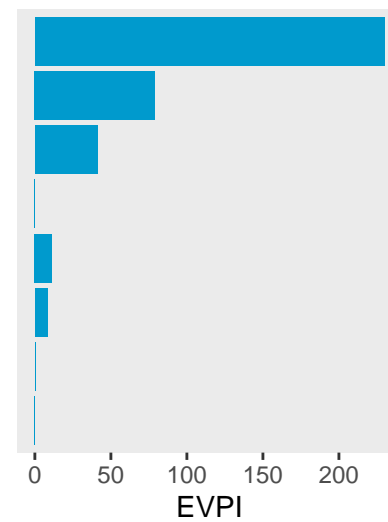


The single graph below is generated using the `cowplot` package (Wilke 2019) to show the outcomes for both the sensitivity analysis (left) and Value of Information analysis (right).

Variable Importance



Value of Information



References

- Luedeling, Eike, Lutz Goehring, and Katja Schiffrers. 2019. *DecisionSupport: Quantitative Support of Decision Making Under Uncertainty*. <https://CRAN.R-project.org/package=decisionSupport>.
- R Core Team. 2019. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. <https://www.R-project.org/>.
- Wickham, Hadley, Winston Chang, Lionel Henry, Thomas Lin Pedersen, Kohske Takahashi, Claus Wilke, Kara Woo, Hiroaki Yutani, and Dewey Dunnington. 2020. *Ggplot2: Create Elegant Data Visualisations Using the Grammar of Graphics*. <https://CRAN.R-project.org/package=ggplot2>.
- Wilke, Claus O. 2019. *Cowplot: Streamlined Plot Theme and Plot Annotations for 'Ggplot2'*. <https://CRAN.R-project.org/package=cowplot>.