**Pseudo-code for FNNR ABM**

Version: 0616

Version description:

1. **Create new module *Household-*** ***demography***
2. *min\_req\_labor* **requirement removed**
3. **Add new input variable: ind ID**

**Model Initiation**

**The *Create-environment* function adopts existing households and land parcel shapefiles to create the environment for modelling**

Create-Environment [

Read in household and land parcel location data, and assign environmental and geographic data pixels accordingly;

]

**The *Create-agent* function reads in attributes data to initialize the model agents.**

Create-Agents [

[will set community-attributes for all community agents later on]

**Set** household-attributes for all household agents (household agents are nested in community agents):

**Set** household ID, Ad village, Natural village, resident location, Charcoal consumption, total rice paddle, total GTGP rice paddle, total dry land, total GTGP dry land, GTGP income

**Set** if-NCFP flag for each household;

**Initialize** total number of out-migrates for each household;

\*\* more attributes will be added later on \*\*

**Set** individual-attributes for all individual agents (individual agents are nested in household agents):

**Set** ind ID, age, gender, education, working status for each individual (each individual is trackable to the household he/she belongs to);

**Set** land parcel-attributes for all household agents:

**Set** indicator of GTGP/non GTGP land for each land parcel;

**Set** area of land for each land parcel;

**Set** plant type for each land parcel;

\*\* more attributes will be added later on \*\*

**Set** PES policy agents:

Set compensation per unit for GTGP;

\*\* more attributes will be added later on \*\*

\*\* all time dependent attributes are set to the state of year 2000 \*\*

]

**The *Initialize-parameters* function presets values to all global parameters**

**\*\*** the name of all read-in variables are space delineated (e.g. GTGP income), while all global parameters which is updating for each time tick, are named with “\_” delineated (e.g. *GTGP\_comp)*\*\*

Initialize-parameters

[

Preset starting values:

Preset *GTGP\_coef* = 0.1;

Preset *mig\_prob* = 0.5;

Preset *comp\_sign* = 0.1;

Preset num\_mig = 0; \* set to 0 for now, will calculate with true data

Preset hh\_empty = N;*\_* \_

Generate a random # (5000-20000) for total\_hh\_income; \* set random for now, will calculate with true data

Preset *GTGP\_comp =* GTGP income;

]

**Major process**

**The *Main-Loop* functionis the main loop in the model. It determines the order in which events occur in the model. The loop runs through time steps 1, 2,…, N (N is the simulation time span in years).**

Main-Loop

[

Call ***Create-environment* function;**

Call ***Create-agent* function;**

Call ***Initialize-parameters* function;**

Call ***Household-*** ***demography* function;**

Loop through all households:

[

If (exist at least one hh member)

Then

[

Calculate *num\_labor (age 15-59);*

Update GTGP compensation, store the value to *GTGP\_comp*;

If (GTGP\_coef\* GTGP\_part > *mig\_prob* AND *GTGP\_comp/* total\_hh\_income *> comp\_sign*)

[

*num\_labor* – 1; \* migration happens \*

*num\_mig* + 1;

]

]

Else

[

hh\_empty = Y;

exit Main-Loop

]

*GTGP\_part\_flag* = 1; \* set flag to enrollment of more land \*

\*\*more logic tests involving personal traits will be added later\*

]

Loop through all land parcels

[

If (*GTGP\_part\_flag* = 1) then

[

Set the GTGP status to GTGP for one non-GTGP that is least close to the household;

]

]

]

**The *Household-*** ***demography* function simulate birth, death, marriage status and education changes of household members**

\*\* divorce and no- birth after spouse death are not included in the model \*\*

Household-demography

[

Preset *birth\_rate* = 0.1;

Preset *birth\_interval* = 2;

Preset *death\_rate* = 0.1;

Preset *marriage\_rate* = 0.1;

Preset *match\_prob* = 0.05;

Make a single\_male\_list;

Loop through all individual agents:

[

If (a female with marriage = 0 AND age >20 AND random #>

*marriage\_rate >* random #) then

[

loop through single\_male\_list:

[

If (random # < *match\_prob*) then

[

Set marriage status to 1 for the female; \* married

Remove the female from the household;

Add the female to her husband’s household;

Assign new ind ID to the married female;

Remove the husband from single\_male\_list;

]

]

]

If (a female with marriage = 1 AND 55< age AND random #>

*birth\_rate* *>* random # AND current\_time - last\_birth\_time > *birth\_interval*) then

[

create an new agent: \*give birth one at a time

[ set age = 0;

set agent = random(1,0);

set education = 0;

assign new ind ID within the household;

assign working status = 6; \*not working

]

]

If (an individual with age > 65 AND random # < *death\_rate*) then

[

Remove the died agent from the household;

]

If (an individual with 19 < age > 7) then

[

assign working status = 5; \*student

education + 1;

]

]

]