**Pseudo-code for FNNR ABM**

Version: 0705-framework-added

Version description:

1. Add new components to the out-migration loop in ***Out-migration* function**: income\_local\_offfarm, non\_GTGP land\_per\_labor, *farm\_work, remittances*. Coefficients for other variable in the function changed accordingly.
2. Calculate initial values for above variables with formula and read-in data highlighted.
3. Set hypothetical *remittances* for new migrants in the future, set *remittances* to zero for the returned migrants, in ***Out-migration* function.**

**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, Rice\_paddles\_mu, Dry\_land\_mu, Rice\_paddles\_GTGP\_mu , Dry\_land\_GTGP\_mu, Restaurant\_cash\_last\_yr, Lodging\_cash\_last\_yr, Transporation\_cash\_last\_yr, Sales\_cash\_last\_yr, Other\_business\_cash\_last\_yr;

**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*, *marriage*, *working\_status,* *remittances(from Yuan\_last\_yr\_remittances\_mig)* 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:

PresetGTGP\_coef = 0.1;

Preset *mig\_prob* = 0.5;

Presetcomp\_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;

Make an out\_migrants\_list with their attribute (*ind-ID*, *age*, *gender*, *education*, *marriage, mig\_years*); \*\* read from an external table

]

**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;**

Call ***Out-migration* function*;***

Call ***GTGP-participation;***

Call ***GTGP-policy;***

Loop through all households:

[

If (exist at least one hh member)

Then

[

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

Calculate non\_GTGP land\_per\_labor

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 simulates 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 *date\_rate* = 0.1;

Preset *marriage\_flag* = 0;

Preset *match\_prob* = 0.05;

Preset *immi\_marriage\_rate* = 0.03;

Make a *single\_male\_list*;

Loop through all individual agents:

[

*marriage\_flag* = 0;

If (a female with marriage = 0 AND age >20 AND random # < *date\_rate*) then

[

loop through *single\_male\_list*:

[

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

[

*marriage\_flag* = 1*;*

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

Set *marriage* to 1 for the male \*\* married

Remove the female from the household;

Assign new ind ID to the married female; \*\* to her husband’s household

Remove the husband from single\_male\_list;

]

]

\*\* If function below test for immigrates through marriage \*\*

If (*marriage\_flag* = 1 AND random # < *immi\_marriage\_rate* ) then

[

Create a new female agents with Age follows N(22.1,2.6) but truncated at 20, education N(8.7,1.3);

Randomly select a male from *single\_male\_list;*

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

Set *marriage* to 1 for the male; \* married

Assign new ind ID to the married female; \*\* to her husband’s household

Assign *working\_status* for the female = 1; \*\* work on own farm

]

]

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

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

[

Record *last\_birth\_time*; for the female

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

[

Set *age* = 0;

Set *gender* = random(1,0);

Set *education* = 0;

set *marriage = 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; \*5 for student

*education* + 1;

]

*age* +1;

]

]

**The *Out-migration* function simulates activities of both out-migration and return-migration for individuals**

\*\* Households are allow one out-migrant at one time tick; while multiple return migrants in single household is allowed \*\*

Out-migration

[

income\_local\_offfarm = sum(Restaurant\_cash\_last\_yr, Lodging\_cash\_last\_yr, Transporation\_cash\_last\_yr, Sales\_cash\_last\_yr, Other\_business\_cash\_last\_yr);

non\_GTGP land\_per\_labor = (Rice\_paddles\_mu + Dry\_land\_mu - Rice\_paddles\_GTGP\_mu - Dry\_land\_GTGP\_mu)/num\_labor;

*remittances* = Yuan\_last\_yr\_remittances\_mig;

*farm\_work = 1 if(working\_status* = 1), otherwise *farm\_work* = 0;

\*\*the loop below simulates return migration\*\*

re\_*mig\_prob* = 0;

Loop through *out\_migrants\_list*:

[

re\_*mig\_prob* = exp(-1.2+0.06\**age*-0.08\**yr\_mig*)/( -1.2+0.06\**age*-0.08\**yr\_mig*))

\*\* *age* in this loop is defined as the age at the time of migration

\*\* the probability is calculated with a generalized linear function

If (random #<re\_*mig\_prob*) then

[

*age* + 1;

Add the individual to the original household;

Set *working\_status =* 1*; \*\** work on own farm after return

Set *remittances* = 0; \*\*no remittances after return

Remove the individual from *out\_migrants\_list;*

]

*age* + 1;

*mig\_years* + 1;

]

\*\*the loop below simulates out-migration\*\*

Loop through all household agents:

[

mig\_flag = 0;

*mig\_prob* = 0;

Loop through all individual agents:

[

If (mig\_flag = 1) then [exit this loop]; \*\* one out-migrant a time

Convert *working\_status to farm\_work* (1,2 in *working\_status* as 1 in *farm\_work,* otherwise 0 in *farm\_work) for this person;*

*mig\_prob* = exp(2.07-0.00015\*income\_local\_offfarm + 0.67\* num\_labor+ 4.36\* Migration\_network - 0.58\* non\_GTGP land\_per\_labor + 0.27\* GTGP\_participation-0.13\* *age* + 0.07\* *gender*+ 0.17\* *education*+ 0.88\* *marriage* + 1.39\**farm\_work +* + 0.001\**remittances* )/(1+ exp(2.07-0.00015\*income\_local\_offfarm + 0.67\* num\_labor+ 4.36\* Migration\_network - 0.58\* non\_GTGP land\_per\_labor + 0.27\* GTGP\_participation-0.13\* *age* + 0.07\* *gender*+ 0.17\* *education*+ 0.88\* *marriage* + 1.39\**farm\_work +* + 0.001\*r*emittances*))

\*\* the probability is calculated with a generalized linear function\*\*

\*\*GTGP weight will be adjusted to a higher value to reflect immediate boosting effect of GTGP to out-migration\*\*

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

[

mig\_flag = 1;

Remove the individual from the household;

\*\*reserve the ind\_ID for the out-migrants in case for return mig

Add the individual to *out\_migrants\_list;*

*working\_status* of the individual changed to 4 \*\* off farm work

*remittances* of the individual set to = a random # follows N(1200,400^2); \*\* set remittances for new migrants

]

]

]

]

**The *GTGP-participation* function**

\*\* Once in GTGP, no exit, until contract expires (simulation pauses and parameters reset)\*\*

GTGP-participation

[

Loop through all non-GTGP land parcels agents:

[

A logistics function will be used to calculate the probability of GTGP participation.

Proposed independent variables: age, gender, education of head of hh; plant\_type, distance\_to\_hh, GTGP\_net\_cash on parcel level; num\_labor on hh level.

\*\* GTGP\_net\_cash = compensation - plant\_type\*unit\_price

\*\* A value for the minimum non-GTGP land will be set to reflect that some parcels farmer are much less likely to enrolled in GTGP

If (GTGP) Then

[

Remove the parcel from non-GTGP land parcels agents;

Add the parcel to GTGP land parcels agents;

]

]

]