**Pseudo-code for FNNR ABM**

Version: 0726

Version description:

1. Some unused variable removed
2. Change some variables to adapt 2014 data
3. Add a minimum household restriction to out-migration loop

**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 [

**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,, Age\_1 (head of hh), Gender\_1, Education\_1, income\_local\_offfarm;

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

**Set** *ind\_ID*, *age*, *gender*, *education*, *marriage*, *working\_status,* for each individual (each individual is trackable to the household he/she belongs to);

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

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

**Set** area\_of\_land for each land parcel;

**Set** plant\_type (Plant\_before\_GTGP or Plant\_last\_nonGTGP )for each land parcel;

**Set** land\_output (Outpu\_before\_GTGP or Output\_nonGTGP) for each land

parcel

**Set** land\_type (Type\_before\_GTGP or Type\_nonGTGP) for each land

parcel;

**Set** time\_land (Travel\_time\_GTGP or Travel\_time\_nonGTGP) for each land

parcel

]

**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:

*\_* \_

Make an out\_migrants\_list with their attribute (*ind\_ID (linked to household ID), Mig\_age , Mig\_gender, Mig\_marriage , Mig\_education ,, Mig\_year, Mig\_remittances*); \*\* read the migrants attributes from the data file \*\*

Preset unit price for crops:

type\_1 = 0.7;

type\_2 = 0.8;

type\_3 = 0.9;

type\_4 = 2.3;

type\_5 = 0;

other\_type = 1;

]

**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-policy;***

Call ***GTGP-participation;***

]

**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

[

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

*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 that have at least 2 members:

[

mig\_flag = 0;

*mig\_prob* = 0;

Loop through all individual agents whose age>15:

[

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;*

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

*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 with all personal migrant attributes in specified in function* ***Initialize-parameters****;*

]

]

]

]

**The *GTGP-participation* function**

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

GTGP-participation

[Preset minimum\_non-GTGP = 0.3; \*\*minimum area of non-GTGP land each household should hold, meet what observed and handle issues of land scarcity \*\*

Loop through all households:

[

Loop through all non-GTGP land parcels agents for a household:

[

Calculate total\_non-GTGP; \*\* add up all non\_GTGP land \*\*

Calculate hh\_size; \*\* add up all household members\*\*

Out of this loop if (total\_non-GTGP < minimum\_non-GTGP);

GTGP\_par\_prob = 0;

crop\_income = land\_output \* unit\_price(with reference to plant\_type)；

Comp\_amt = area\_of\_land \* unit\_comp;

GTGP\_net\_cash = Comp\_amt - crop\_income; \*\* on parcel level

\*\*A logistics function will be used to calculate the probability of GTGP participation on parcel level \*\*

GTGP\_par\_prob = exp(2.52 - 0.012\* Age\_1 - 0.29\* Gender\_1 + 0.01\* Education\_1 + 0.001\* hh\_size - 2.45\*land\_type + 0.0006\* GTGP\_net\_cash + 0.04\* time\_land)/(1 + exp(2.52 - 0.012\* Age\_1 - 0.29\* Gender\_1 + 0.01\* Education\_1 + 0.001\* hh\_size - 2.45\*land\_type + 0.0006\* GTGP\_net\_cash + 0.04\* time\_land));

If (random #<GTGP\_par\_prob) then

[

Remove the parcel from non-GTGP land parcels agents;

Add the parcel to GTGP land parcels agents;

]

Age\_1 + 1; \*\* age of hh head increment

]

]

**The *GTGP-policy* function**

GTGP-policy

[

Scenario\_1: unit\_comp = 270;

Scenario\_2: unit\_comp = 135;

Scenario\_3: for rice\_puddy, unit\_comp = 270;

for dry\_land, unit\_comp = 135;

Scenario\_4: first 4 year unit\_comp = 800;

after 4 year unit\_comp = 200;

]