TOC

[Package dependencies: 2](#_Toc91263004)

[Modules: 2](#_Toc91263005)

[Required Data: 2](#_Toc91263006)

[Farmer 2](#_Toc91263007)

[Industry 3](#_Toc91263008)

[Community 4](#_Toc91263009)

[Government 4](#_Toc91263010)

[GIS files 4](#_Toc91263011)

[Some notes on data prepration 5](#_Toc91263012)

[Agent behaviors 5](#_Toc91263013)

[Farmer 5](#_Toc91263014)

[candidate\_refinery 7](#_Toc91263015)

[Refinery 7](#_Toc91263016)

[Community 8](#_Toc91263017)

[Consumer 9](#_Toc91263018)

[Government 9](#_Toc91263019)

[Model details 10](#_Toc91263020)

[Model flow chart 10](#_Toc91263021)

[Farmer decision rules 10](#_Toc91263022)

[Industry decision rules 11](#_Toc91263023)

[Community decision rules 14](#_Toc91263024)

[Consumer decision rules 16](#_Toc91263025)

[Government decision rules 17](#_Toc91263026)

[Code numbers 18](#_Toc91263027)

[Refinery type (tech\_type) 18](#_Toc91263028)

[Land use 18](#_Toc91263029)

[Feedstock 18](#_Toc91263030)

# Package dependencies:

* biorefineries (need to be installed from file)
* mortgage
* scipy
* geopandas
* cv2

# Modules:

* daycent\_data\_processing.py – a module for transforming the Daycent simulation data into response matrices for ABM
* parameters.py – a module storing the tunable parameters in the ABM (note some parameters in csv files and not stored in this module are also tunable parameters)
* config.py – an interface to load data and parameters (except for government related parameters)
* government.py – a module for all government related parameters
* gen\_agents.py – a module to generate agents
* agents.py – a module for defining agents Class and their functions
* general\_methods\_physical.py – a module defining functions relating to physical processes in the ABM
* general\_methods\_economic.py – a module defining functions relating to economic processes in the ABM
* main.py – the main function for ABM, simulates the agent decisions and interactions
* cal\_stats.py – a module that defines all data post-processing functions for the ABM, including the visualization functions
* visualization\_new.py – a module that draws all figures from the ABM simulation

# Required Data:

## Farmer

* farmer\_attributes.csv – a table showing the attributes of all farmer agents
* farmer\_ini\_state.csv – a table showing the initial conditions of all farmer agents
* farmer\_cost\_table.csv – a table showing all costs related to farmer agents
* fertilizer\_table.csv – a table showing the fertilizer requirements for all crops considered
* patch\_attributes.csv – a table showing the attributes of all land patches
* patch\_ini\_state.csv – a table showing the initial conditions of all land patches
* patch\_carbon\_table\_x.csv – a table showing the soil carbon response matrix for all land patches
* patch\_N\_loads\_x.csv – a table showing the N release response matrix for all land patches
* patch\_water\_use\_x.csv – a table showing the water use response matrix for all land patches
* patch\_yield\_table\_x.csv – a table showing the crop yield response matrix for all land patches
* perennial\_yield\_adj\_table.csv – a table showing the crop yield adjustment factor during the start and end of perennial grass life cycle
* adjacency\_matrix.csv – a matrix specifying the neighborhood relationship between farmers
* neighbor\_influence\_matrix.csv – a matrix specifying the intensity of neighborhood influence

## Industry

* can\_refinery\_attributes.csv – a table showing the attributes of all candidate biorefinery plants (including corn and cellulosic)
* can\_BCHP\_attributes.csv – a table showing the attributes of all candidate BCHPs
* can\_cofire\_attributes.csv – a table showing the attributes of all candidate cofire power plants
* ini\_ref\_attributes.csv – a table showing the attributes of all initial biorefineries
* ref\_farmer\_dist\_matrix.csv – a matrix specifying the distances between farmers and the candidate biorefinery plants
* BCHP\_farmer\_dist\_matrix.csv – a matrix specifying the distances between farmers and candidate BCHP locations
* cofire\_farmer\_dist\_matrix.csv – a matrix specifying the distances between farmers and candidate cofire power plant locations
* corn\_ref\_farmer\_dist\_matrix.csv – a matrix specifying the distances between farmers and the initial corn biorefinery plants
* ref\_ref\_dist\_matrix.csv – a matrix specifying the distances between biorefineries
* refinery\_feedstock\_table.csv – a table showing the feedstock requirements of different biofacilities
* refinery\_product\_yield\_table.csv – a table showing the bioproduct yields of all feedstocks for biorefineries
* biofacility\_product\_yield\_table.csv – a table of the product yields for BCHP and cofire power plants
* ref\_cost\_table.csv – a table showing the costs related to biorefineries
* ref\_job\_table.csv – a table showing the jobs created by biorefineries
* refinery\_water\_use\_table.csv – a table showing the water use requirement by biorefineries

## Community

* community\_attributes.csv – a table showing the attributes of all community agents
* community\_ini\_states.csv – a table showing the initial conditions of all community agents

## Government

* gov\_ref\_tax\_subsidy.csv – a table showing the subsidy provided by the government for supporting cellulosic biofacilities
* gov\_related\_prices.csv – a table showing the prices that is used for government decision making
* RFS\_volume.csv – a table showing the RFS cellulosic biofuel production mandate
* external\_provided\_prices.csv – a table showing the prices of all feedstocks and bioproducts

## GIS files

* farms\_sagamon\_farm1000\_with\_urban.shp – the farmers and their land patches
* farms\_sagamon\_farm1000\_with\_urban\_selected.shp – marginal land boundaries
* can\_industry\_locs.shp – the candidate locations of biorefineries and biofacilities

## Some notes on data prepration

* The farmer land patch data is from Yan and Roy (2016) (Yan, L., Roy, D.P., 2016. Conterminous United States crop field size quantification from multi-temporal Landsat data. Remote Sens. Environ. 172, 67–86.) dataset. The original data is raster file, we can turn it into shapefile and then using clustering to group adjacent land parcels into land patches and farmers.
* The various distance datasets are generated based on highway data through GIS network analysis
* The response matrixes are generated from Daycent simulation using daycent\_data\_processing.py.
* The locations of refineries and BCHPs are randomly generated; the locations of co-firing plants are generated based on existing coal power plant dataset (included in the ‘insight\_model\data\GIS’ folder, PowerPlants\_US\_202004.shp).

# Agent behaviors

## Farmer

* **Attributes**: discount factor, risk factor, farm size, information source (info\_use), benefit factor (Bayesien Network, BN variable), concern factor (BN variable), marginal land availability (lql, BN variable), environmental concern (PT\_imp\_env, BN variable), adopt\_priority (the priority of growing perennial grass), loss factor, learning rate, price update rate, community ID, farmer type (type), sensitivity to community environmental attitude (sensi\_community), farmer’s rate of learning from experience (tao\_factor), neighbors, neighbor weights, patch ID, patch areas, patch CRP eligibility, patch slope, patch soil loss factor
* **States:** prior belief of crop prices (price\_prior), posterior belief of crop prices (price\_post), received price, land use, contract, age of perennial grasses, a state showing if perennial grass installation has been failed (failed), crop yield, N release from land, carbon sequestration, fertilizer use, revenue, patch TMDL eligibility, patch BCAP availability, importance of environment (imp\_env, BN variable), attitude toward environment (environ\_sen), neighborhood adoption condition (peer\_ec), familiarity with perennial grass (max\_fam), willingness to grow perennial grass (SC\_Will), portion of land growing perennial grass (SC\_Ratio), a state variable showing if farmer adopts perennial grass (is\_adopt), a factor representing farmer’s realization of the long-term economic benefits of perennial grass (learning\_factor)
* **Methods**
  + forecasts\_price: forecast the prices of crops
  + cal\_peer\_ec: calculate the portion of neighbor that is growing perennial grass
  + cal\_PT\_imp\_env: calculate the probability table of imp\_env (refer to the BN model)
  + update\_environ\_sen: update farmer’s sensitivity to environment at each time step
  + update\_crop\_no\_physical\_model: calculate crop yield, N release, fertilizer use and perennial grass age
  + update\_profit: calculate the profit at each time step
  + cal\_utility\_annual\_crop: calculate the utility function of annual crops (corn, soybean)
  + cal\_utility\_perennial: calculate the utility function of perennial grass
  + cal\_max\_potential\_U\_crop: calculate the maximum utility among corn and soybean
  + cal\_max\_potential\_U\_peren: calculate the maximum utility among different types of perennial grasses
  + cal\_max\_potential\_profit\_peren: calculate the maximum profit among different types of perennial grasses
  + cal\_adopt\_priority: calculate the adoption priority among different patches
  + identify\_contract\_patches: identify the patches that make contract with industry
  + cal\_peren\_break\_even\_price: calculate the break even price of perennial grass
  + cal\_min\_contract\_price: calculate the minimum acceptable perennial grass price for making contract
  + land\_use\_decision\_econo: land use decisions if decided on economic rules, used to facilitate farmer’s ‘real’ land use decisions
  + land\_use\_decision\_bn: land use decision based on BN
  + update\_attitude\_self: update farmer’s attitude toward perennial grass at each time step (not used with the current model configuration)
  + learn\_from\_experience: function for farmer to update their perception of longterm economic benefits of perennial grass from their own experience
  + learning\_from\_neighbor: function for farmer to update their perception of longterm economic benefits of perennial grass under neighbor influences
  + update\_AT\_neighbor: update environmental attitude based on neighbor influence
  + bn\_inference: function to perform BN inferencing, for assisting land\_use\_decision\_bn calculation
  + compile\_farmer\_contract: function to combine farmer contract land use decisions

## candidate\_refinery

The biorefineries and biofacilities under planning stage

* **Attributes**: location ID, community ID, refinery type, capacity, minimum required IRR, whether it is a co-op (co\_op), distance to farmers
* **States:** expected water use (WU), expected NPV, expected IRR, feedstock requirement, investment cost, feedstock availability, adjusted investment cost (after subsidy), expected interest payment each year for the investment cost (interest\_payment), average distance to farmer (aver\_dist), fixed operation cost, viable operation cost
* **Methods**
  + cal\_supply\_curve: calculate the local supply curve (amount v.s. price) of feedstocks to the refinery
  + check\_supply\_curve: check the supply curve to match the feedstock amount and the corresponding price
  + cal\_NPV: calculate the NPV and IRR of the plant investment
  + cal\_water\_use: calculate the water requirement of the plant

## Refinery

* **Attributes**: location ID, feedstock requirement (feedstock\_amount), capacity, technology type, tax reduction condition, tax rate, subsidy, viable operation cost, fixed operation cost, maximum allowable deficit in operation (max\_deficit), investment cost (invest), interest payment, co-op, distances to farmer, the amount of bagasse, the year of construction, average distance to farmer (aver\_dist)
* **States**: purchased feedstocks, sold feedstocks, transportation distances of purchased feedstocks (purchased\_feed\_dist), production year since construction (production\_year), accepted subsidy, profit, water use (WU), biofuel production, byproduct production, contracted feedstock amount (contracted\_patch\_amount, this is an expected value in making the contract), contracted feedstock prices, actually supplied contracted feedstock (contracted\_patch\_supply), transportation distances of contracted feedstocks, patch IDs that made contracts with the plant (contracted\_patch\_ID), farmer IDs that made contracts with the plant (contracted\_farmer\_ID)
* **Methods**:
  + cal\_PBE: calculate the break-even feedstock price of the biorefinery or biofacility, presented in refinery gate price
  + produce\_biofuel: for the biorefinery or biofacility to calculate the biofuel and byproduct production based on the obtained feedstock
  + feed\_management\_after\_contract: decide the amounts of feedstocks to buy or sell based on the received contracted feedstocks
  + cal\_profit: calculate the profit of biorefinery or biofacility at each time step
  + cal\_stop\_production: identify if the refinery or biofacility termination condition is reached
  + pick\_contracts: select the contracts from the potential contracts offered by farmer
  + is\_continue\_contract: identify if the refinery or biofacility still needs to make more contracts with farmer to secure feedstock supply
  + make\_contracts: identify the contract and contract prices with the farmers
  + check\_contract\_continuity: check if there are contracts reached their maximum duration

## Community

* **Attributes**: the threshold willingness to accept new biorefinery investment (accept\_threshold), increasing rate of environmental sensitivity (base\_increase\_environ), maximum environmental sensitivity (max\_attitude), sensitivity of environmental attitude toward N release, portion of farmer in the community (ratio\_farmer), N release limit, maximum allowable land use change (max\_LU), satisfying level of job opportunity provided by biorefinery (max\_job), average historical N release (average\_N\_loads)
* **States**: attitude toward biorefinery investment (attitude), agricultural revenue (revenue), water availability, N release, a state variable showing if the community denied a biorefinery investment
* **Methods**:
  + ini\_community: initiate the community agent
  + cal\_N\_load\_in\_community: calculate the N release in the community
  + cal\_revenue\_in\_community: calculate the agricultural revenue generated in the community
  + cal\_attitude: update the community’s attitude toward environment at each time step
  + cal\_willingess: the community’s willingness to support a biorefinery investment
  + pred\_lU\_change: calculate the estimated land use change caused by the cellulosic biorefinery within the community

## Consumer

* **Attributes**: increasing rate of willingness to pay (IRW), maximum willingness to pay (max\_WP)
* **States**: willingness to pay primum for cellulosic biofuel, ethanol price
* **Methods**
  + cal\_willingness\_to\_pay: update the consumer’s willingness to pay for primum for cellulosic biofuel
  + cal\_ethanol\_price: update the cellulosic ethanol price based on willingness to pay

## Government

* **Attributes**: TMDL cap for watershed N release (TMDL), slope threshold for identifying land patches eligible for TMDL subsidy (slope\_limits), N release threshold for identifying land patches eligible for TMDL subsidy (TMDL\_N\_limits), scaling\_factor (not used)
* **States**: price of cellulosic wavier credit (CWC\_price), volume of renewable fuel standard cellulosic biofuel production mandate (RFS\_volume), adjusted cellulosic biofuel price based on CWC\_price (RFS\_adjusted\_cell\_ethanol\_price), the slope threshold selected by gov agent (slope\_limit\_ID), the N release threshold selected by gov agent (N\_limit\_ID)
* **Methods**
  + update\_RFS: adjust the RFS cellulosic mandated amount based on actual cellulosic biofuel production
  + cal\_IRR\_adj\_factor: determine if the minimum IRR requirement by industry agent will be adjusted because of RFS mandate change
  + cal\_CWC\_price: calculate the price of cellulosic wavier credit
  + cal\_cell\_ethanol\_price: calculate the price of cellulosic biofuel based on CWC price
  + update\_TMDL: update the N release limit for identifying the land patches eligible for TMDL subsidy

# Model details

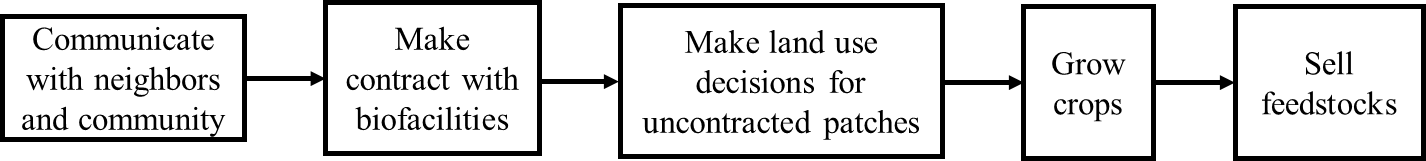
## Model flow chart

Diagram

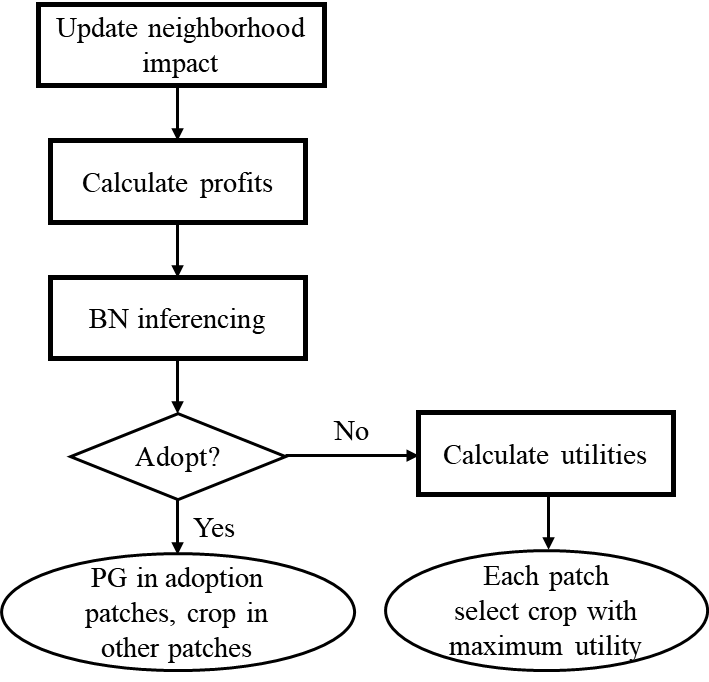
Description automatically generated

## Farmer decision rules

* Decision process in a time step



* Perennial grass adoption rule



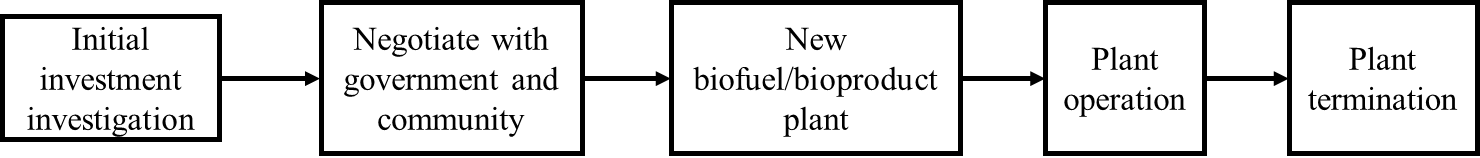
where *U* is the utility of a crop, *P* is the profit of a crop ($/ha), is the risk aversion parameter, is the loss aversion parameters, is the discount factor, *p* is the price of the crop ($/ton), *Yt* is the yield of the crop (ton/ha) at year *t*, *St* is the subsidy provided for the crop ($/ton) at year *t*, *Ct* is the cost of the crop ($/ha) at year *t*, is the neighborhood related factor that relates to the farmer’s value of the environmental service provided by the crop, *EV* is the environmental service provided by the crop ($/ha), *R* is the land rent ($/ha), *IC* is the installation cost ($/ha), *T* is the time horizon (the life span of Miscanthus, 15 years).

If the profit of perennial grass is lower than 300 $/acre, the perennial grass adoption will be determined through the BN model trained from the survey data; otherwise, the farmer agent will adopt perennial grass instantly.

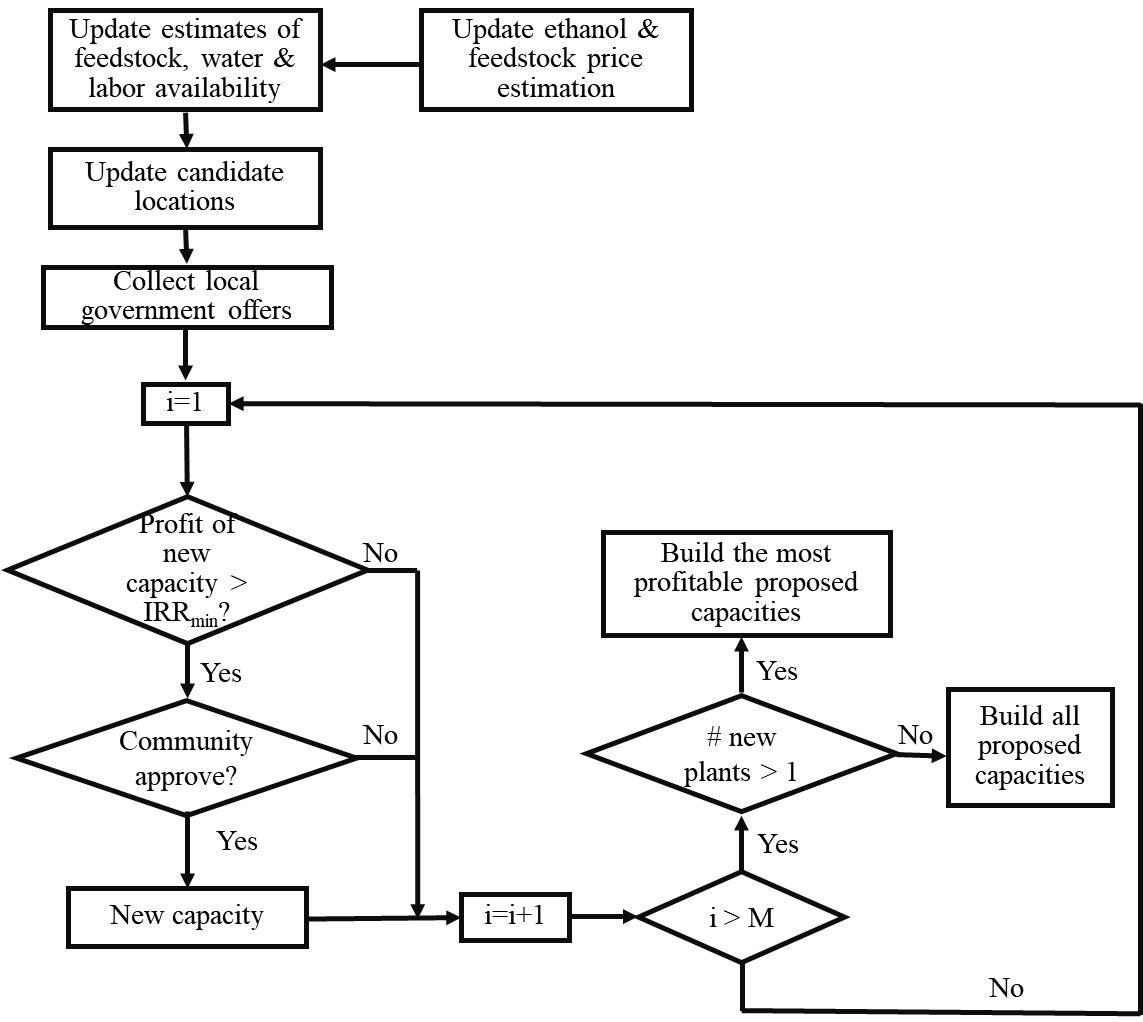
* Parameters
  + land\_rent: in parameters.py
  + marginal\_land\_rent\_adj: determines the marginal land rent as a ratio of land\_rent, in parameters.py
  + learning\_factor: indicates the possible levels of , can be increased if the farmer is growing perennial grass or if they have neighbors growing perennial grass (learn\_from\_experience and learning\_from\_neighbor)

## Industry decision rules

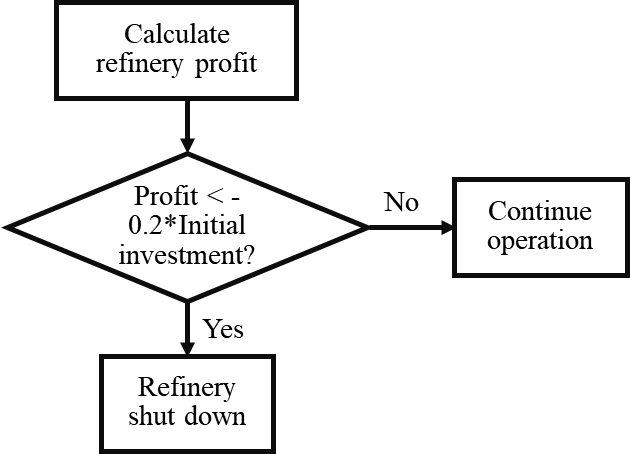
* Decision process in a time step



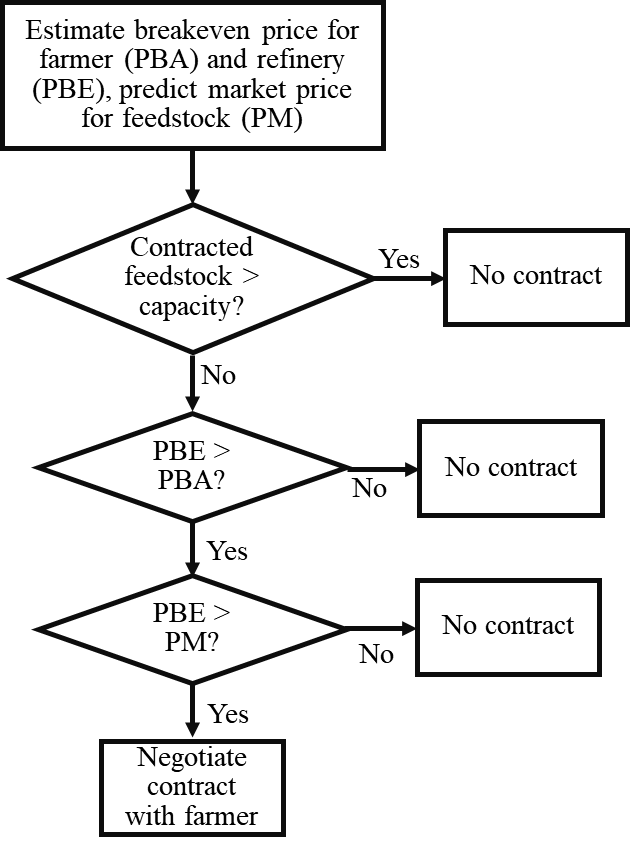
* Behavior rules
  + Investing rule



* + Termination rule



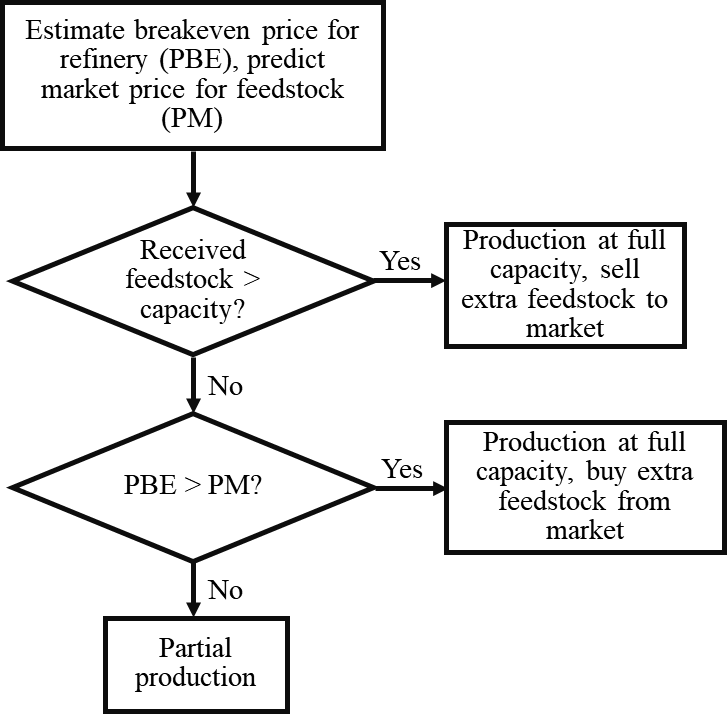
* + Contracting rule



The contracted feedstock price for each farmer varies and is determined by a lower bound *LB*, an upper bound *UB*, and the market tightness factor :

Here the upper bound *UB* is determined as the breakeven price for bio-facility (*P*BE) and BEPAM price (*P*BEPAM), *UB=*min(*P*BE, *P*BEPAM+ *C*T *dmax*), where *C*T is the feedstock transportation cost, and *d*max is the maximum transportation distance assumed by the modeler; and the lower bound *LB* = *C*T *d* + *C*S + max(*P*BA + *P*M,cell), where *d* the distance between farmer and bio-facility, *C*S the feedstock storage cost, *P*BA the breakeven price for the farmer, and *P*M,cell the cellulosic feedstock market price. Therefore, a farmer closer to bio-facility will receive a higher contract price for cellulosic feedstock. If *UB* is smaller than *LB*, there will be no contract between the farmer and the bio-facility.

* + Buying feedstock rule



Refineries and supply chain use the following learning-by-doing rule to update their costs as they have more experiences:

where *RCt*+1 is the refinery cost at time step t+1, *CA* is the refinery capacity, is the baseline refinery capacity, is the learning by doing updating rate

* Parameters
  + learn\_by\_do\_rate: the updating factor of learning-by-doing, in parameters.py
  + base\_cap\_for\_learn\_by\_do: the baseline refinery capacity for learning-by-doing, in parameters.py
  + base\_feed\_amount\_for\_learn\_by\_do: the baseline supply chain handling amount for learning-by-doing, in parameters.py
  + allowable\_deficit: in parameters.py

## Community decision rules

* Behavior rules

Diagram

Description automatically generated

where revenue is farmer’s *revenue* in the community, *#Jobs* is the number of job opportunity provided by refinery, *CA* is refinery capacity, *WU* is the refinery water use, is the expected land use change by refinery, *XAM* and *XWU* are the current agricultural revenue and currently available water resources respectively, the maximum allowable land use change allowed by the community. and are the percentages of agricultural and non-agricultural population in the community.

Community’s attitude to environment:

where is the community’s attitude to environment, is the maximum level of , *l* is the community’s sensitivity to N release, *TMDL* is the maximum N release allowed by the TMDL policy, is the base increase rate of community’s attitude to environment.

* Parameters
  + accept\_threshold: the maximum value *w* below which community would accept new biorefinery, in community\_attributes.csv
  + base\_increase\_rate: , in community\_attributes.csv
  + max\_attitude: , in community\_attributes.csv
  + sensi\_N: *l*, in community\_attributes.csv
  + ratio\_farmer: , in community\_attributes.csv
  + N\_limit: *TMDL*, in community\_attributes.csv
  + max\_LU\_change: , in community\_attributes.csv
  + max\_jobs: *#Jobs*max, in community\_attributes.csv
  + max\_capacity: , in community\_attributes.csv

## Consumer decision rules

* Behavior rules

where is the modeled ethanol price from BEPAM, and is the increase of ethanol price as a result of increased willingness to pay.

where is the initial willingness to pay, is the maximum willingness to pay, is the increase rate of willingness to pay with time *t*.

* Parameters
  + ini\_WP: , in parameters.py
  + IRW: , in parameters.py
  + max\_WP: , in parameters.py

## Government decision rules

* Behavior rules
  + RFS

If cellulosic biofuel production < RFS mandate, cellulosic wavier credit (CWC) will be implemented, with price:

where *IF* is the inflation factor with respect to 2009, is the petroleum price.

If cellulosic biofuel production < 0.5 \* RFS mandate, meanwhile government agent allows adjustment of RFS mandate, RFS mandate will be adjusted to 1.4 \* cellulosic biofuel production

* BCAP

Provide installation cost share and biomass subsidy to farmers that provide biomass to cellulosic biofacitlities.

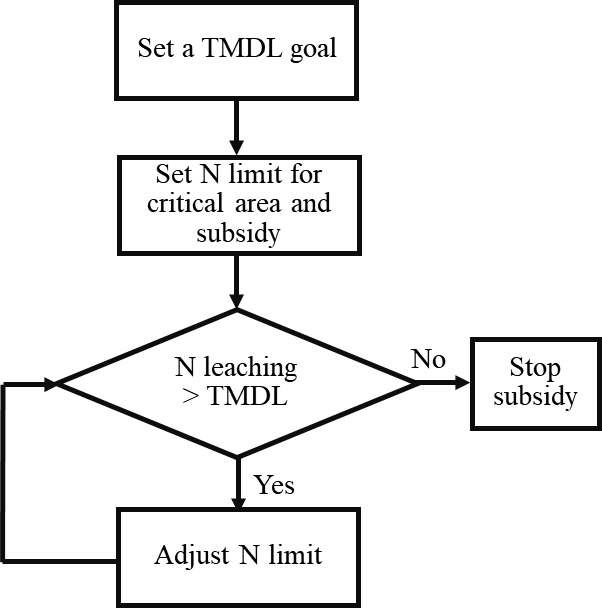
* Subsidy to biofacilities

Tax reduction, investment cost sharing, and product subsidy

* CRP

Provide subsidy ($/ha) to farmers that do not grow row crops in their land

* TMDL



* Parameters
  + Maintain\_RFS: whether the government allows RFS mandate to be adjusted, in parameters.py
  + CRP\_subsidy: in parameters.py
  + CRP\_relax: whether the government allows CRP land to harvest perennial grass, in parameters.py
  + TMDL\_subsidy: in parameters.py
  + BCAP\_subsidy: in parameters.py
  + BCAP\_cost\_share: in parameters.py
  + tax\_reduction: in parameters.py
  + tax\_rate: in parameters.py
  + carbon\_price: price of carbon credit, in parameters.py
  + nitrogen\_price: price of N credit, in parameters.py
  + refinery\_ cost\_share: in gov\_ref\_tax\_subsidy.csv
  + production\_subsidy: product subsidy for refinery, in gov\_ref\_tax\_subsidy.csv

# Code numbers

## Refinery type (tech\_type)

1 – corn refinery, 2 – cellulosic ethanol refinery, 3 – cellulosic biodiesel refinery, 4 – co-production of biodiesel and ethanol from lipid cellulosic feedstock, 5 – co-fire power plant with 5% limit of cellulosic feedstock, 6 – co-fire power plant with 15% limit of cellulosic feedstock, 7 – biomass combined heat and power system

## Land use

1 – corn, 2 – soy, 3 – mis, 4 – switch, 5 – sorghum, 6 – cane, 7 – fallow, 8 – CRP, 30 – new miscanthus installation, 40 – new switchgrass installation

## Feedstock

0 – corn grain, 1 – soy grain, 2 – corn stover, 3 – miscanthus, 4 – switchgrass, 5 – bagasse, 6 – oil sorghum, 7 – lipid cane, 8 – sorghum for ethanol-biodiesel coproduction, 9 – lipid cane for ethanol-biodiesel coproduction