An Agent-Based Examination into the Effect of Information Flow on Electricity Consumption

Adam Leising, Stanford University
Zhang Ting, Xi'an Jiaotong University

September 2006

Abstract

An agent-based model has been developed to explore the effect of de-centralized versus centralized information diffusion on the consumption of dirty and clean electricity. It is found that the effectiveness and robustness of an environmentalist's influence is significantly increased by de-centralized information flow, such as that occurring within a small neighborhood. This finding is somewhat counter-intuitive, for as a de-centralized information topology is defined here the information flow is limited to the confines of the neighborhood. This finding suggests that de-centralized information diffusion may have significant cost-saving capabilities.

****

This paper is a product of the Santa Fe Institute's Beijing 2006 Complex Systems Summer School, and is therefore not only an examination of a relevant topic but also an exercise in a method relatively new to the authors, which in this case is agent-based modeling.

The topic examined in this paper is the effect of information transfer among neighbors on the consumption of either dirty or clean electricity. The information transfer portion of the model deals with the variable beta, which is the price premium a consumer is willing to pay for clean electricity. The remainder of the model is essentially economical, with functions that govern the evolution of price and demand.

Each consumer is defined as either an environmentalist, a centrist, or an economist and assigned an initial demand quantity (from a normal distribution) as well as an initial beta. An environmentalist is one who always, no matter the price, chooses clean electricity (in this model the choice between dirty and clean electricity is all or none). An economist is one who always, no matter the social pressure, chooses the cheaper electricity. A centrist is one who may choose clean electricity if the price premium is less than his beta value (of course, if clean electricity is less expensive than dirty electricity, all consumers will choose clean).

Initial prices for dirty and clean electricity are chosen. Based on a consumer's beta value, he will choose dirty or clean, thereby affecting the market demand for dirty and clean electricity. A change in market demand causes a change in market price, creating a feedback loop between demand and price. At the same time, consumers are communicating with each other and influencing each other's beta values.

Once the model reaches equilibrium, which is defined as a miniscule change in market demand from one run to the next, data on the number of consumers choosing each type of electricity, the market demand for and price of each type of electricity, and the distribution of beta are presented.

Information Transfer

The change in a consumer's beta value is a function of the difference between his beta value and that of his neighbor as well as the distance between him and his neighbor. The function is as follows:

$$\beta_i(k+1) = \beta_i(k) + [\Sigma (\delta_i(\beta_j - \beta_i) / 2D_{ij})] / N$$

with $\delta_i = 1 - [(d_i - d_{min}) / (d_{max} - d_{min})]$
$d_i = abs (\beta_j - \beta_i)$
$D_{ij} = sqrt [(x_j - x_i)^2 + (y_j - y_i)^2]$.

The summation is over j, the relevant neighbors; N is the number of these relevant neighbors. Delta is a normalization factor used to weigh the influence of a similar beta value more than the influence of a less similar beta value; d_{min} and d_{max} are selected from the relevant neighbors. The 2 in the denominator prevents over-convergence.

This function is not biased toward an increase or a decrease in beta; when two centrists interact, the change in beta experienced by each centrist is equal in magnitude but opposite in direction. Hence, in the absence of environmentalists and economists, who change centrists' beta without having their own changed, the distribution of beta converges on the initial mean.

The beta value for economists is zero, and this does not change. The appropriate beta value for environmentalists is, however, not so obvious. It may be argued that for a consumer who is committed to choosing clean over dirty electricity, the appropriate beta value is infinity. Yet this value is not practical mathematically. It may also be the case that an environmentalist's stated beta value far exceeds his actual beta value. The beta values for environmentalists are, therefore, based on revealed preference, as follows:

$$\beta_{envr} = [(P_c / P_d) - 1]1.1,$$

with P_c being the price of clean electricity and P_d the price of dirty electricity. The factor of 1.1 means that beta for an environmentalist is 10% greater than it need be for the adjusted price of clean electricity to always be less than the price of dirty electricity. The beta value for an environmentalist is based on the initial prices of dirty and clean electricity and changes with the changing prices, though it is not allowed to decrease.

This method of calculating beta for the environmentalists is conservative and allows for the effect of an economist to be more powerful than the effect of an environmentalist. The significance of this shortcoming may be explored by varying the 1.1 factor.

Two topologies are employed in the model, centralized and de-centralized. In the centralized topology, each consumer communicates with every other consumer, essentially making every consumer a neighbor of everyone. The effect of this is that information is very diffuse. In the de-centralized topology, neighborhoods have been created in which the neighbors communicate among themselves only, and not with other neighborhoods. Information in the de-centralized topology is densely local. Presumably, the de-centralized topology would be more susceptible to trends, be they good or bad. Clarification of the effect of each of these topologies on consumer behavior is one of the objectives of this project.

Economic Behavior

The first function governing economic behavior pertains to the price elasticity of demand and is as follows:

$$d_i(k+1) = d_i(k)[1 + \varepsilon_D(P(k+1) / P(k) - 1)]$$

with d_i being quantity demanded and ε_D the price elasticity of demand (approximately -0.2 for electricity in the short-term [1]). Each time a consumer experiences a change in price, he adjusts his consumption according to the above function. This adjustment occurs even if a consumer willingly switches from cheaper dirty electricity to more expensive clean electricity. It may be the case that when a consumer willingly incurs the higher cost of clean electricity, he does not reduce his consumption of electricity; presumably, this behavior varies from consumer to consumer. In this model, however, the situation is simplified and consumers always react to changes in price. Future models may deal with this uncertainty in a more sophisticated fashion.

The second function governing economic behavior pertains to the change in price caused by a change in demand, and is as follows:

$$P(k+1) = P(k)[1 + \varepsilon_I(D(k+1) / D(k) - 1)]$$

with ε_I being the inverse elasticity, the percent change in price given a percent change in demand. This function applies to both types of electricity, just with different subscripts.

Dirty electricity is considered a mature technology with a developed market such that an increase in demand for dirty electricity places pressure on the fuel market, leading to increased costs and increased prices. The inverse elasticity for dirty electricity is therefore positive in sign (values range from 0.5 to 2.5 in the long-term and may be larger in the short-term [2]).

Clean electricity, on the other hand, is considered a developing technology yet to exhaust the benefits of economies of scale and learning by doing. It is posited that these benefits outweigh the potential detriments of increased demand. Indeed, the photovoltaic industry has experienced a decrease in costs, which in a competitive business environment translates into a decrease in price, of approximately 20% for every doubling of cumulative production [3]. This indicates an inverse elasticity for clean electricity of approximately -0.2.

It may be noted that the two inverse elasticities have opposite signs, with that for dirty electricity being positive and that for clean electricity being negative. This creates the possibility of either a balancing (negative) feedback loop or a reinforcing (positive) feedback loop. To illustrate, consider an increase in the market demand for clean electricity, which has two effects. On the one hand, an increase in the market demand for clean electricity leads to a decrease in its price, which leads to an increase in its market demand, thereby forming a reinforcing feedback loop. On the other hand, an increase in the market demand for clean electricity causes a decrease in the market demand for dirty electricity (assuming approximately constant total (dirty plus clean) market demand), leading to a decrease in its price and then an increase in its demand, which finally causes a decrease in the market demand for clean electricity, completing a balancing feedback loop.

The magnitudes of the feedback loops are largely controlled by the ratio of the two inverse elasticities. A large epsilon inverse magnitude for dirty electricity relative to the epsilon inverse magnitude for clean electricity, for example, creates a balancing feedback loop that lends resilience to the market demand for dirty electricity such that the market demand for clean electricity increases but does not surpass that of dirty electricity (compare Figure 1, for which epsilon inverse for dirty electricity is 2.00 and that for clean electricity is -0.20, to Figure 2, for which epsilon inverse for dirty electricity is 0.50 and that for clean electricity is -0.20).

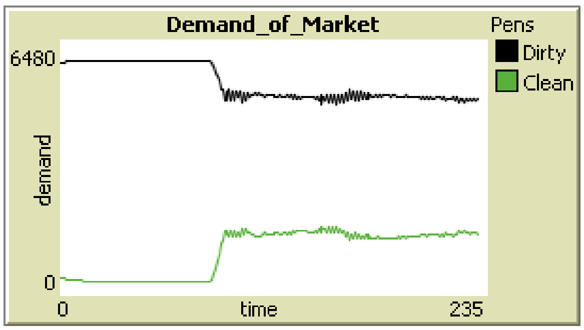

Figure 1: Market Demand for Dirty and Clean Electricity with ε_I dirty = 2.00

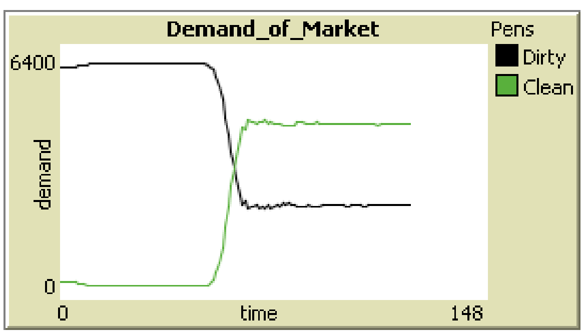

Figure 2: Market Demand for Dirty and Clean Electricity with ε_I dirty = 0.50

Results

It was found that the influence of environmentalists is significantly more powerful in the de-centralized topology than in the centralized topology. With a centralized topology and 45% of the consumers identified as environmentalists, the market demand for clean electricity surpasses that of dirty electricity at approximately the 80th iteration. With a de-centralized topology, however, only 3% of the consumers need be environmentalists for the market demand for clean electricity to surpass that of dirty electricity at approximately the 80th iteration (refer to Figures 3 and 4).

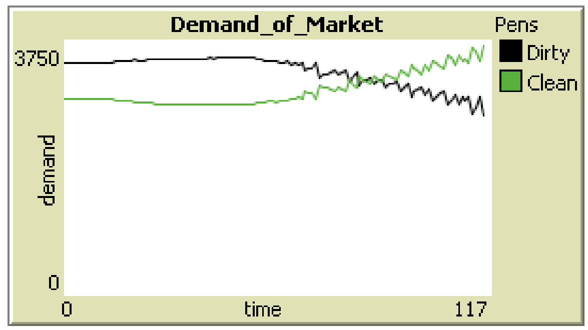

Figure 3: Market Demand for Dirty and Clean Electricity with Centralized Topology and Envr = 45%

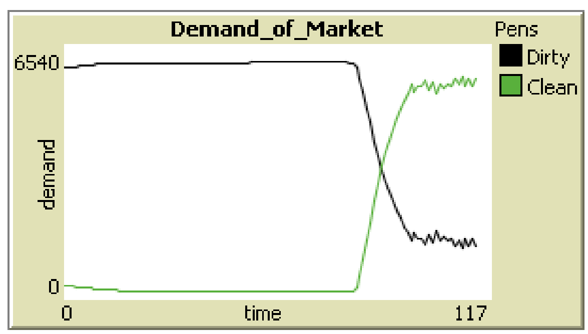

Figure 4: Market Demand for Dirty and Clean Electricity with De-centralized Topology and Envr = 3%

Environmentalists have two effects: as a result of their commitment to purchasing clean electricity over dirty electricity, they prevent the price of clean electricity from rising exceedingly due to a lack of demand; secondly, they continually influence their neighbors, increasing their beta values and therefore making them more likely to switch to clean electricity. The first effect is the same in a centralized topology as it is in a de-centralized topology, for this effect is simply a function of the number of environmentalists. It is the influence of environmentalists on their neighbors, therefore, that is greater in the de-centralized topology than in the centralized topology.

It was also found that the influence of environmentalists is more robust in the de-centralized topology than in the centralized topology. In the de-centralized topology with 3% environmentalists, if an equal amount of economists are added, the market demand for clean electricity persists in surpassing that of dirty electricity, while an analogous scenario for the centralized topology results in a negation of the environmentalists' influence, at least within a comparable timeframe (refer to Figures 5 and 6).

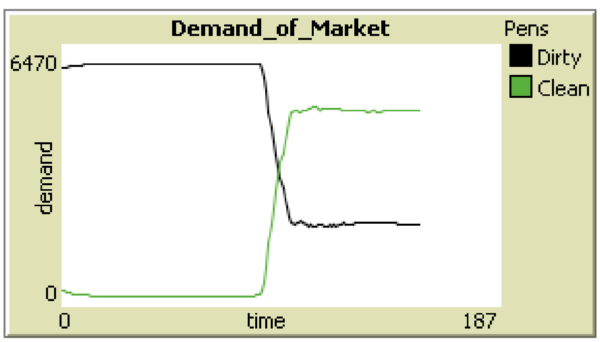

Figure 5: Market Demand for Dirty and Clean Electricity with De-centralized Topology and Envr = 3%, Econ = 3%

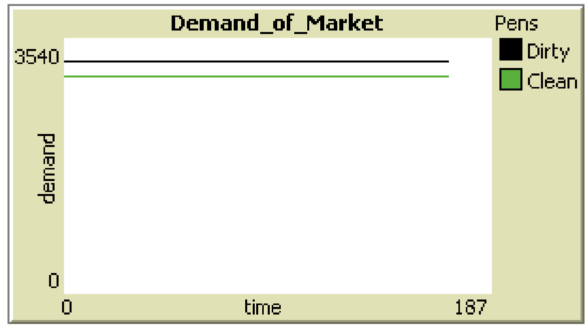

Figure 6: Market Demand for Dirty and Clean Electricity with Centralized Topology and Envr = 45%, Econ = 45%

Conclusion

This project is the result of an endeavor into a field relatively unfamiliar to the authors. There are many improvements to be made, the least of which will be made possible by access to computers more powerful than personal laptops.

It has been determined that the effectiveness and robustness of an environmentalist's influence is significantly increased by de-centralized information flow, such as that occurring within a small neighborhood. This finding suggests that de-centralized information diffusion may have significant cost-saving capabilities, for in the presence of a centralized information topology, greater resources would be necessary for a comparable impact.

****

The authors would like to thank the Santa Fe Institute and the McGee and Rudolph Organizations for their support, without which this project would not have been possible.

Contact information for Adam Leising: aleising@stanford.edu, and for Zhang Ting: tzhang08@gmail.com.

References

1. Stevens, B. and L. Lerner, California Energy Commission. 1996. *Testimony on the Effect of Restructuring on Price Elasticities of Demand and Supply: Prepared for the August 14, 1996 ER 96 Committee Hearing*.

2. Wiser, R., Bolinger, M. and M. St. Clair. 2005. *Easing the Natural Gas Crisis: Reducing Natural Gas Prices through Increased Deployment of Renewable Energy and Energy Efficiency*. LBNL-56756. Berkeley, Calif.: Lawrence Berkeley National Laboratory.

3. Handleman, C. 2001. *An Experience Curve Based Model for the Projection of PV Module Costs and Its Policy Implications*.
Utility Photovoltaic Experience Conference and Exhibition 2001, Sacramento, California.

Appendix: NetLogo Code

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; define the global variables ;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
globals
[
times         ;; simulation times
time          ;; simulation clock
flag
nearby        ;; store the neighbors of a consumer
Demand_dirty  ;; sum of demand of dirty energy
Demand_clean  ;; sum of demand of clean energy
Price_dirty   ;; price of dirty energy
Price_clean   ;; price of clean energy
]
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; define the consumer's attributes  ;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
patches-own
[
  Comfortable? ;; whether the consumer is a comfortable consumer? true for comfortable
consumer, false for constained consumer
  Category     ;; the category of the consumer. "Centrist; Economist; Environmentalist"
  Beta         ;; beta
  Price        ;; the price that consumer want to pay
  Choose       ;; whether use clean or dirty energy? "Dirty" for dirty energy, "Clean" for clean
energy
  Demand_initial  ;; consumer's initial demand
  Demand       ;; demand
  setnumber
]
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;    reset                ;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
to reset
ca
end
;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;
;;        setup         ;;
;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;
to setup
  ifelse times = 0
```

```
  [ ca
  output-type "times  "
  output-type "P_dirty  "
  output-type "P_clean   "
  output-type "Demand_dirty    "
  output-type "Demand_clean  "
  output-type "#comf  "
  output-type "#cons  "
  output-type "Dirty user  "
  output-type "Clean user  "
  output-print "stop_time"
  ]
  [ cp
    ct
    clear-all-plots
  ]
  set time 0  ;; simulation step start from 0
  set flag false
  set Price_dirty Price_dirty_initial
  set Price_clean Price_clean_initial
  ;; setup whether the consumer is a comf consumer
  setup-comf
  ;; setup the category and the beta of the consumer
  setup-category-beta
  ;; setup the demand of the consumer
  setup-demand
  ;; calculate the slope of the price-demand function
  setup-market
  ;; initial the plot
  do-plot
end
;;;;;;;;;;;;;;;;;;;;;;;;;;
; initial consumer type  ;
;;;;;;;;;;;;;;;;;;;;;;;;;;
to setup-comf
  ask patches
  [ ifelse random-float 1.0 < percent-comf
      [ set-comfortable ]
      [ set-constrained ]
  ]
end
;; comfortable consumer represented by red color
to set-comfortable
  set Comfortable? true
  set pcolor red
end
```

```
;; constrained consumer represented by black color
to set-constrained
  set Comfortable? false
  set pcolor blue
end
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;initial consumer's category, beta, demand and price;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
to setup-category-beta
  ask patches
  [
    ifelse Comfortable?
    ;; if the agent is a comfortable consumer, to process as below
    [
      let prandom random-float 1.0
      ifelse prandom < percent-of-comf_cent
      [ set Category "Centrist"
        set Beta random-normal beta_mean-of-comf_cent beta_SD-of-comf_cent
      ]
      [ ifelse prandom < percent-of-comf_cent + percent-of-comf_econ
        [ set Category "Economist"
          set Beta 0
        ]
        [ set Category "Environmentalist"
          set Beta ((Price_clean_initial / Price_dirty_initial) - 1) * 1.1
        ]
      ]
    ]
    ;; if the agent is a constained consumer, to process as below
    [
      let prandom random-float 1.0
      ifelse prandom < percent-of-cons_cent
      [ set Category "Centrist"
        set Beta random-normal beta_mean-of-cons_cent beta_SD-of-cons_cent
      ]
      [ ifelse prandom < percent-of-cons_cent + percent-of-cons_econ
        [ set Category "Economist"
          set Beta 0
        ]
        [ set Category "Environmentalist"
          set Beta ((Price_clean_initial / Price_dirty_initial) - 1) * 1.1
        ]
      ]
    ]
    ;;decide the price and which kind of energy
    ifelse Price_dirty * ( 1 + Beta) < Price_clean
```

```
   [ set Price Price_dirty
     set Choose "Dirty"
     sprout 1 [ set color black
                set shape "star" ]  ;; make a turtle to represent different kind of energy: black turtle
for dirty energy, green turtle for clean energy
   ]
   [ set Price Price_clean
     set Choose "Clean"
     sprout 1 [ set color green
                set shape "star" ]
   ]
 ]
end
;;;;;;;;;;;;;;;;;;;;;;;;;;;
;initial consumer's demand;
;;;;;;;;;;;;;;;;;;;;;;;;;;;
to setup-demand
 ask patches
 [
  set Demand_initial random-normal 10  2 ;; Demand_initial obeys the distribution of N(10,2)
  if Demand_initial < 0 [ set Demand_initial 0 ]  ;; Demand_initial shouldn't be less than 0
  ifelse Comfortable?
   [ set Demand Demand_initial ]
   [ set Demand Demand_initial * alpha]
 ]
end
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;  setup market demand parameters of price-demand function   ;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
to setup-market
set Demand_dirty 0
set Demand_clean 0
ask patches[
 ifelse Choose = "Dirty"
  [set Demand_dirty Demand_dirty + Demand ]
  [set Demand_clean Demand_clean + Demand ]
 ]
end
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;                    run the simulation                  ;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
to go
 ;; wait for user to stop drawing: when pressing the mouse button, simulation stops
 if mouse-down?
```

```
  [ stop ]
;; simulation time plus one
set time time + 1
;; decide the neighborhood of different system type
ifelse system-type = "centralized"
  [ set nearby moore-offsets 12 ]
  ;; set setnumber of de-centralized system for consumers
  [ ask patches
    [ set setnumber (floor (pxcor / 5)) + (floor (pycor / 5 ))* 5
    ]
  ]
;; decide consumer's type-whether he is comfortable
decide-comfortable
;; update consumer's Beta, Price and Demand
consumer-decision
;; calculate the total demand of dirty and clean energy
calculate-market
;; update the plot
do-plot
if times < simulation_times and ( flag = true )
[ set times times + 1
  output-type times
  output-type "     "
  output-write Price_dirty
  output-type "     "
  output-write Price_clean
  output-type "    "
  output-write Demand_dirty
  output-type "    "
  output-write Demand_clean
  output-type "     "
  output-write count patches with [Comfortable? = true]
  output-type "     "
  output-write count patches with [Comfortable? = false]
  output-type "     "
  output-write count patches with [Choose = "Dirty"]
  output-type "     "
  output-write count patches with [Choose = "Clean"]
  output-type "     "
  output-print time
  setup
]
if times >= simulation_times
[
set times times - 1
stop
```

```
  ]
end
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;        simulate just for one round       ;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
to go_one_round
  if mouse-down?
    [ stop ]
  set time time + 1
  ifelse system-type = "centralized"
    [ set nearby moore-offsets 12 ]
    ;; set setnumber of de-centralized system for consumers
    [ ask patches
      [ set setnumber (floor (pxcor / 5)) + (floor (pycor / 5 ))* 5
      ]
    ]
  decide-comfortable
  consumer-decision
  calculate-market
  do-plot
  if flag = true
  [ output-type times
    output-type "     "
    output-write Price_dirty
    output-type "     "
    output-write Price_clean
    output-type "    "
    output-write Demand_dirty
    output-type "    "
    output-write Demand_clean
    output-type "     "
    output-write count patches with [Comfortable? = true]
    output-type "     "
    output-write count patches with [Comfortable? = false]
    output-type "     "
    output-write count patches with [Choose = "Dirty"]
    output-type "     "
    output-write count patches with [Choose = "Clean"]
    output-type "     "
    output-print time
  ]
  if flag = true
  [stop]
end
to-report moore-offsets [ n ]
   let dim n * 2 + 1
```

```
  let result n-values (dim ^ 2)
         [list (? mod dim - n)
              (floor (? / dim) - n)]
 report remove [0 0] result
end
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; consumers update their beta, demand and price ;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
to consumer-decision
ask patches
  [
  let effect-neighbor 0
  let effect-neighbor-temp 0
  ifelse system-type = "centralized"
    [set effect-neighbor patches at-points nearby]
    [set effect-neighbor-temp patches with [setnumber = setnumber-of myself]
    ;;remove the consumer itself from the effect-neighbor
    set effect-neighbor effect-neighbor-temp with [ not (pxcor-of self = pxcor-of myself and
pycor-of self = pycor-of myself) ]
    ]
  ;; decide the Beta
  ;; calculate the sum effect of beta value from the neighbors for a consumer
  let effect-of-beta 0
  let average-of-effect 0
  let d_min min values-from effect-neighbor [ abs ( Beta-of self - Beta-of myself) ]
  let d_max max values-from effect-neighbor [ abs ( Beta-of self - Beta-of myself) ]
  if count effect-neighbor >= 1
     ;;if effect-neighbor is an agentset, process as below (use an agentset function values-from)
  [
    set effect-of-beta sum ( values-from effect-neighbor
                       [  ( 1 - ( (abs (Beta-of self - Beta-of myself) - d_min) / ( 1 + d_max - d_min ) ) )
* (Beta-of self - Beta-of myself ) /
                       ( 2 * sqrt ( (pxcor-of self - pxcor-of myself ) ^ 2 + (pycor-of self - pycor-of
myself) ^ 2) )
                       ] )
    set average-of-effect effect-of-beta / count effect-neighbor
  ]
  ;; update Beta
  ifelse Category = "Centrist"
    [ set Beta Beta + average-of-effect ]
    [ ifelse Category = "Economist"
      [ set Beta Beta ]
      [ ifelse Beta >= ( (price_clean / price_dirty) - 1) * 1.1
        [ set Beta Beta ]
        [ set Beta ( (price_clean / price_dirty) - 1) * 1.1 ]
      ]
```

```
    ]
  ;; decide the new price, put it to a new temp variable, bacause we need former Price to calculate
the Demand
  let Price_temp 0
  ifelse Price_dirty * ( 1 + Beta) < Price_clean
  [ ifelse Choose = "Clean"               ;; the probability of change Choose is 0.1
    [ ifelse random-float 1.0 < 0.1
      [ set Price_temp Price_dirty
        set Choose "Dirty"
        ask turtles-here [ set color black ]
      ]
      [ set Price_temp Price_clean ]
    ]
    [ set Price_temp Price_dirty ]
  ]
  [ ifelse Price_dirty * ( 1 + Beta ) > Price_clean
    [ ifelse Choose = "Dirty"
      [ ifelse random-float 1.0 < 0.1
        [ set Price_temp Price_clean
          set Choose "Clean"
          ask turtles-here [ set color green ]
        ]
        [ set Price_temp Price_dirty ]
      ]
      [ set Price_temp Price_clean ]
    ]
    [ set Price_temp Price
      set Choose Choose ]
  ]
  ;; update Demand
  ifelse Comfortable?
  [ set Demand Demand * ( 1 + epsilon * ( ( Price_temp / Price ) - 1 ) )]
  [ set Demand alpha * Demand_initial ]
  ;; update Price
  set Price Price_temp
  ]
end
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;   update the consumers' type   ;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
to decide-comfortable
ask patches [
let effect-neighbor 0
  ifelse system-type = "centralized"
    [set effect-neighbor patches at-points nearby]
    [set effect-neighbor patches with [setnumber = setnumber-of myself]]
```

```
 ifelse Comfortable?
 [ if count effect-neighbor with [ Comfortable? = false ] > 0.5 * count effect-neighbor
   [ if random-float 1.0 < 1
     [ set-constrained ]
   ]
 ]
 [ if count effect-neighbor with [ Comfortable? = true] > 0.5 * count effect-neighbor
   [ if random-float 1.0 < 1
     [ set-comfortable ]
   ]
 ]
]
end
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;calculat total demand fo dirty and clean energy;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
to calculate-market
let Demand_dirty_temp 0
let Demand_clean_temp 0
let Price_dirty_temp 0
let Price_clean_temp 0
ask patches[
 ifelse Choose = "Dirty"
   [set Demand_dirty_temp Demand_dirty_temp + Demand ]
   [set Demand_clean_temp Demand_clean_temp + Demand ]
 ]
ifelse Demand_dirty_temp <= 0 or Demand_clean_temp <= 0 or ( time > 200 and ( abs
( Demand_dirty_temp - Demand_dirty ) < 0.00000001 and abs ( Demand_clean_temp -
Demand_clean ) < 0.00000001 ) )
[
 set Price_dirty_temp Price_dirty
 set Price_clean_temp Price_clean
 set flag true
]
[
 set Price_dirty_temp Price_dirty * ( 1 + epsilon_inverse * ( (Demand_dirty_temp /
Demand_dirty) - 1) )
 set Price_clean_temp Price_clean * ( 1 + epsilon_x * ( (Demand_clean_temp / Demand_clean)
- 1) )
]
set Price_dirty Price_dirty_temp
set Price_clean Price_clean_temp
set Demand_dirty Demand_dirty_temp
set Demand_clean Demand_clean_temp
```

```
end
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;          draw plots        ;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
to do-plot
set-current-plot "Consumer type"
set-current-plot-pen "Comfortable"
plot count patches with [ Comfortable? = true ]
set-current-plot-pen "Constrained"
plot count patches with [ Comfortable? = false ]
set-current-plot "Demand_of_market"
set-current-plot-pen "Dirty"
plot Demand_dirty
set-current-plot-pen "Clean"
plot Demand_clean
set-current-plot "Price"
set-current-plot-pen "Dirty"
plot Price_dirty
set-current-plot-pen "Clean"
plot Price_clean
set-current-plot "Choose"
set-current-plot-pen "Dirty"
plot count patches with [ Choose = "Dirty" ]
set-current-plot-pen "Clean"
plot count patches with [ Choose = "Clean" ]
set-current-plot "Beta"
set-current-plot-pen "Beta_mean"
plot mean values-from patches [Beta]
set-current-plot-pen "Beta_SD"
plot standard-deviation values-from patches [Beta]
set-current-plot "Demand"
set-current-plot-pen "Demand_mean"
plot mean values-from patches [Demand]
set-current-plot-pen "Demand_SD"
plot standard-deviation values-from patches [Demand]
end
```