

附錄 A 捷運沿線土地使用變遷模式之 NetLogo 模擬程式碼

```
breeds [  
  stations  
  rails  
  newstation  
  newrails  
]  
globals [  
  maxlanduse  
  maxpop  
]  
patches-own [  
  landuse ; landuse type  
  population ; patch's pop.  
  capacity  
  roadw  
  interchange  
  value  
  increase-value  
  rail  
  ca-rail  
  around  
  ca-around  
  dis1 ; the distance to closest MRT station  
  dis2  
  dis3  
  dis4  
  dis5  
  resident ; counts how many neighboring cells are residential areas  
  commerce ; counts how many neighboring cells are commercial areas  
  industry ; counts how many neighboring cells are industrial area  
  agriculture ; counts how many neighboring cells are agricultrial area  
  agr  
  ind  
  res  
  com  
  magr  
  mind  
  mres  
  mcom  
  amagr  
  amind  
  amres  
  amcom  
  p-agriculture  
  p-industry  
  p-resident  
  p-commerce
```



```
t          ; years
]
turtles-own [home-pos]
```

to setup

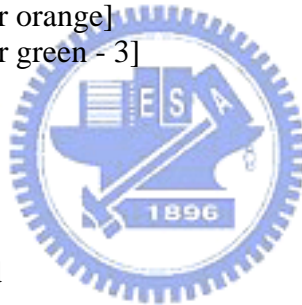
```
clear-patches
set maxlanduse 12
file-open "landuse.txt"
  ask patches [
    set landuse file-read
    if landuse = 0 [set pcolor white]
    if landuse = 1 [set pcolor green]
    if landuse = 2 [cell-resident]
    if landuse = 3 [set pcolor brown]
    if landuse = 4 [cell-commerce]
    if landuse = 5 [set pcolor green + 1]
    if landuse = 6 [set pcolor sky]
    if landuse = 7 [set pcolor black]
    if landuse = 8 [set pcolor magenta]
    if landuse = 9 [set pcolor blue]
    if landuse = 10 [set pcolor cyan]
    if landuse = 11 [set pcolor orange]
    if landuse = 12 [set pcolor green - 3]
  ]
file-close
```

```
file-open "population.txt"
  ask patches [
    set population file-read
  ]
file-close
```

```
file-open "capacity.txt"
  ask patches [
    set capacity file-read
  ]
file-close
```

```
file-open "roadw.txt"
  ask patches [
    set roadw file-read
  ]
file-close
```

```
file-open "value.txt"
  ask patches [
    set value file-read
  ]
file-close
```



```
file-open "increase-value.txt"
  ask patches [
    set increase-value file-read
  ]
file-close
```

```
file-open "rail.txt"
  ask patches [
    set rail file-read
  ]
file-close
```

```
file-open "around.txt"
  ask patches [
    set around file-read
  ]
file-close
```

```
ask patches
  [ set agriculture count neighbors with [landuse = 1]
    set resident count neighbors with [landuse = 2]
    set industry count neighbors with [landuse = 3]
    set commerce count neighbors with [landuse = 4]
    set ca-rail count neighbors with [rail = 1]
    set ca-around count neighbors with [around = 1]
  ]
```

```
clear-turtles
create-stations
ask stations
[ set home-pos list xcor ycor ]
ask stations
[ setxy (item 0 home-pos) (item 1 home-pos)
set shape "house ranch"
set size 1
]
```

```
if show-mrt? [create-rails]
ask rails
[ set home-pos list xcor ycor ]
ask rails
[ setxy (item 0 home-pos) (item 1 home-pos)
set shape "rail"
]
```

```
if new-stations [create-newstation]
ask newstation
[ set home-pos list xcor ycor ]
ask newstation
```



```

[ setxy (item 0 home-pos) (item 1 home-pos)
set shape "house ranch"
]

if new-mrt [create-newrails]
ask newrails
[ set home-pos list xcor ycor ]
ask newrails
[ setxy (item 0 home-pos) (item 1 home-pos)
set shape "newrail"
]

ask patches [
  set dis1 distancexy 1 28
  set dis2 distancexy 5 10
  set dis3 distancexy 1 -6
  set dis4 distancexy -6 -27
  set dis5 distancexy 8 28
  set interchange distancexy 15 8
]

clear-all-plots
do-plots1
do-plots2
end

to go
ask patches [
  if ca-rail > 1 [if increase-capacity = "+20" [set capacity capacity + 20]]
  if ca-rail > 1 [if increase-capacity = "+40" [set capacity capacity + 40]]
  if ca-rail > 1 [if increase-capacity = "+60" [set capacity capacity + 60]]
  if ca-rail > 1 [if increase-capacity = "+80" [set capacity capacity + 80]]
  if ca-rail > 1 [if increase-capacity = "+100" [set capacity capacity + 100]]
]

ask patches [
  ifelse (dis1 > distance-station / 50) [set dis1 50] [ifelse dis1 > 0 [] [set dis1 1]]
  ifelse (dis2 > distance-station / 50) [set dis2 50] [ifelse dis2 > 0 [] [set dis2 1]]
  ifelse (dis3 > distance-station / 50) [set dis3 50] [ifelse dis3 > 0 [] [set dis3 1]]
  ifelse (dis4 > distance-station / 50) [set dis4 50] [ifelse dis4 > 0 [] [set dis4 1]]
  ifelse (dis5 > distance-station / 50) [set dis5 50] [ifelse dis5 > 0 [] [set dis5 1]]
  ifelse (interchange > 20) [set interchange 100] [ifelse interchange > 0 [] [set
interchange 1]]
]

ask patches [
  set t t + 1 ]

ask patches [
  set value value * (1 + increase-value) * 0.1 ]

```



```

ask patches [
  set population ((max-planning-pop.) - ((pop-growing-rate * 0.01) ^ t) *
(max-planning-pop. - population)) ]

ask patches [
  set magr
  (e ^ (population * agriculture-pop-rate * 0.01 + resident * ca-a-resident +
commerce * ca-a-commerce
+ agriculture * ca-a-agriculture + industry * ca-a-industry + capacity * a-capacity
* 0.01 + roadw * a-roadw + a-rail * ca-rail
+ 1 / dis1 * a-mrt + 1 / dis2 * a-mrt + 1 / dis3 * a-mrt + 1 / dis4 * a-mrt + 1 /
interchange * a-interchange + value * a-value * 0.001 ))
]
ask patches [
  set mind
  (e ^ (population * industry-pop-rate * 0.01 + resident * ca-i-resident +
commerce * ca-i-commerce
+ agriculture * ca-i-agriculture + industry * ca-i-industry + capacity * i-capacity *
0.01 + roadw * i-roadw + i-rail * ca-rail
+ 1 / dis1 * i-mrt + 1 / dis2 * i-mrt + 1 / dis3 * i-mrt + 1 / dis4 * i-mrt + 1 /
interchange * i-interchange + value * i-value * 0.001 ))
]
ask patches [
  set mres
  (e ^ (population * resident-pop-rate * 0.01 + resident * ca-r-resident +
commerce * ca-r-commerce
+ agriculture * ca-r-agriculture + industry * ca-r-industry + capacity * r-capacity *
0.01 + roadw * r-roadw + r-rail * ca-rail
+ 1 / dis1 * r-mrt + 1 / dis2 * r-mrt + 1 / dis3 * r-mrt + 1 / dis4 * r-mrt + 1 /
interchange * r-interchange + value * r-value * 0.001 ))
]
ask patches [
  set mcom
  (e ^ (population * commerce-pop-rate * 0.01 + resident * ca-c-resident +
commerce * ca-c-commerce
+ agriculture * ca-c-agriculture + industry * ca-c-industry + capacity * c-capacity
* 0.01 + roadw * c-roadw + c-rail * ca-rail
+ 1 / dis1 * c-mrt + 1 / dis2 * c-mrt + 1 / dis3 * c-mrt + 1 / dis4 * c-mrt + 1 /
interchange * c-interchange + value * c-value * 0.001 ))
]

ask patches [
  set agr
  (e ^ (population * agriculture-pop-rate * 0.01 + resident * ca-a-resident +
commerce * ca-a-commerce
+ agriculture * ca-a-agriculture + industry * ca-a-industry + capacity * a-capacity
* 0.01 + roadw * a-roadw
- interchange * a-interchange + value * a-value * 0.0001 ))
]

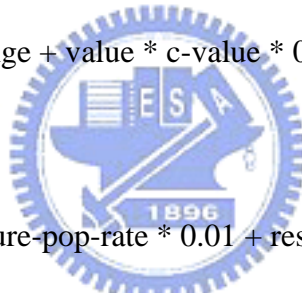
```

```

ask patches [
  set ind
  (e ^ (population * industry-pop-rate * 0.01 + resident * ca-i-resident +
commerce * ca-i-commerce
+ agriculture * ca-i-agriculture + industry * ca-i-industry + capacity * i-capacity *
0.01 + roadw * i-roadw
- interchange * i-interchange + value * i-value * 0.0001 ))
]
ask patches [
  set res
  (e ^ (population * resident-pop-rate * 0.01 + resident * ca-r-resident +
commerce * ca-r-commerce
+ agriculture * ca-r-agriculture + industry * ca-r-industry + capacity * r-capacity *
0.01 + roadw * r-roadw
- interchange * r-interchange + value * r-value * 0.0001 ))
]
ask patches [
  set com
  (e ^ (population * commerce-pop-rate * 0.01 + resident * ca-c-resident +
commerce * ca-c-commerce
+ agriculture * ca-c-agriculture + industry * ca-c-industry + capacity * c-capacity
* 0.01 + roadw * c-roadw
- interchange * c-interchange + value * c-value * 0.0001 ))
]

ask patches [
  set amagr
  (e ^ (population * agriculture-pop-rate * 0.01 + resident * ca-a-resident +
commerce * ca-a-commerce
+ agriculture * ca-a-agriculture + industry * ca-a-industry + capacity * a-capacity
* 0.01 + roadw * a-roadw + a-rail * ca-rail + a-rail * ca-around
+ 1 / dis1 * a-mrt + 1 / dis2 * a-mrt + 1 / dis3 * a-mrt + 1 / dis4 * a-mrt + 1 / dis5
* a-mrt + 1 / interchange * a-interchange + value * a-value * 0.001 ))
]
ask patches [
  set amind
  (e ^ (population * industry-pop-rate * 0.01 + resident * ca-i-resident +
commerce * ca-i-commerce
+ agriculture * ca-i-agriculture + industry * ca-i-industry + capacity * i-capacity *
0.01 + roadw * i-roadw + i-rail * ca-rail + i-rail * ca-around
+ 1 / dis1 * i-mrt + 1 / dis2 * i-mrt + 1 / dis3 * i-mrt + 1 / dis4 * i-mrt + 1 / dis5 *
i-mrt + 1 / interchange * i-interchange + value * i-value * 0.001 ))
]
ask patches [
  set amres
  (e ^ (population * resident-pop-rate * 0.01 + resident * ca-r-resident +
commerce * ca-r-commerce
+ agriculture * ca-r-agriculture + industry * ca-r-industry + capacity * r-capacity *
0.01 + roadw * r-roadw + r-rail * ca-rail + r-rail * ca-around
+ 1 / dis1 * r-mrt + 1 / dis2 * r-mrt + 1 / dis3 * r-mrt + 1 / dis4 * r-mrt + 1 / dis5 *

```



```

r-mrt + 1 / interchange * r-interchange + value * r-value * 0.001 ))
]
ask patches [
  set amcom
  (e ^ (population * commerce-pop-rate * 0.01 + resident * ca-c-resident +
commerce * ca-c-commerce
+ agriculture * ca-c-agriculture + industry * ca-c-industry + capacity * c-capacity
* 0.01 + roadw * c-roadw + c-rail * ca-rail + c-rail * ca-around
+ 1 / dis1 * c-mrt + 1 / dis2 * c-mrt + 1 / dis3 * c-mrt + 1 / dis4 * c-mrt + 1 / dis5
* c-mrt + 1 / interchange * c-interchange + value * c-value * 0.001 ))
]

ask patches
[ifelse around-mrt [set p-agriculture (amagr / (amagr + amind + amres + amcom))]
[ifelse MRT? [set p-agriculture (magr / (magr + mind + mres + mcom))]
[set p-agriculture (agr / (agr + ind + res + com))]]]

ask patches
[ifelse around-mrt [set p-industry (amind / (amagr + amind + amres + amcom))]
[ifelse MRT? [set p-industry (mind / (magr + mind + mres + mcom))]
[set p-industry (ind / (agr + ind + res + com))]]]

ask patches
[ifelse around-mrt [set p-resident (amres / (amagr + amind + amres + amcom))]
[ifelse MRT? [set p-resident (mres / (magr + mind + mres + mcom))]
[set p-resident (res / (agr + ind + res + com))]]]

ask patches
[ifelse around-mrt [set p-commerce (amcom / (amagr + amind + amres + amcom))]
[ifelse MRT? [set p-commerce (mcom / (magr + mind + mres + mcom))]
[set p-commerce (com / (agr + ind + res + com))]]]

ask patches
[ifelse ((count patches with [landuse = 4]) <= (2325 * commercial-use * 0.01)) and
(landuse = 2) and (p-commerce > p-resident)
[cell-commerce] [if (landuse = 4) and (p-resident > (p-commerce)) [cell-resident]]]
ask patches
[ifelse (landuse = 1) and (p-resident > p-agriculture) [cell-resident] []]
ask patches
[ifelse (landuse = 3) and (p-resident > p-industry) [cell-resident] []]
ask patches
[ifelse (landuse = 0) and (p-resident > p-industry) and (p-resident > p-commerce)
and (p-resident > p-agriculture) [cell-resident]
[if (landuse = 0) and ((count patches with [landuse = 4]) <= (2325 * commercial-use
* 0.01)) and (p-commerce > p-agriculture) and
(p-commerce > p-industry) and (p-commerce > p-resident) [cell-commerce]]]

do-plots1
do-plots2
end

```

```

to create-stations
  set-default-shape stations "house ranch"
  ask patches at-points [[1 28] [5 10] [1 -6] [-6 -27] ]
    [ sprout-stations 1 []]
end

to create-rails
  set-default-shape rails "rail"
  ask patches at-points [[-6 -27] [-6 -26] [-6 -25] [-5 -24] [-5 -23] [-5 -22] [-4 -21]
[-4 -20] [-4 -19] [-3 -18]
[-3 -17] [-2 -16] [-2 -15] [-2 -14] [-1 -13] [-1 -12] [-1 -11] [0 -10] [0 -9] [0 -8]
[0 -7] [1 -6] [1 -5] [2 -4]
[2 -3] [2 -2] [3 -1] [3 0] [3 1] [3 2] [4 3] [4 4] [4 5] [4 6] [4 7] [4 8] [5 9] [5 10]
[5 11] [4 12] [4 13] [4 14]
[4 15] [4 16] [3 17] [3 18] [3 19] [3 20] [2 21] [2 22] [2 23] [2 24] [1 25] [1 26]
[1 27] [1 28] [1 29] [0 30]
[0 31] [0 32] [0 33] [-1 34] [-1 35] [-1 36] [-1 37]]
    [ sprout-rails 2 []]
end

to create-newrails
  set-default-shape rails "newrail"
  ask patches at-points [[-15 28] [-14 28] [-13 28] [-12 28] [-11 28] [-10 28] [-9 28]
[-8 28] [-7 28] [-6 28] [-5 28]
[-4 28] [-3 28] [-2 28] [-1 28] [0 28] [1 28] [2 28] [3 28] [4 28] [5 28] [6 28] [7
28] [8 28] [9 29] [10 30] [11 31]
[12 32] [13 33] [13 34] [13 35] [13 36] [14 36] [15 36]]
    [ sprout-rails 2 []]
end

to create-newstation
  set-default-shape newstation "house ranch"
  ask patches at-points [[8 28]]
    [ sprout-newstation 1 []]
end

to cell-commerce
  set landuse 4
  set pcolor red
end

to cell-resident
  set landuse 2
  set pcolor yellow
end

to do-plots1
  set-current-plot "landuse"

```



```

set-current-plot-pen "agriculture"
plot count patches with [landuse = 1]

set-current-plot-pen "resident"
plot count patches with [landuse = 2]

set-current-plot-pen "industry"
plot count patches with [landuse = 3]

set-current-plot-pen "commerce"
plot count patches with [landuse = 4]
end

to do-plots2
  set-current-plot "populations"
  set-current-plot-pen "pops"
  plot ceiling sum values-from patches [population]
end

to-report pop.
  report ceiling sum values-from patches [population]
end

to-report number-vacant
  report count patches with [landuse = 0]
end

to-report number-agriculture
  report count patches with [landuse = 1]
end

to-report number-resident
  report count patches with [landuse = 2]
end

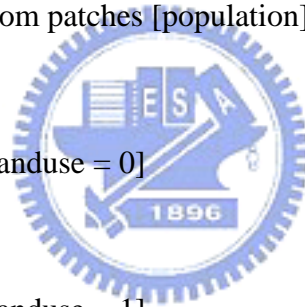
to-report number-industry
  report count patches with [landuse = 3]
end

to-report number-commerce
  report count patches with [landuse = 4]
end

to-report number-green
  report count patches with [landuse = 5]
end

to-report number-organization
  report count patches with [landuse = 6]
end

```



end

to-report number-military

 report count patches with [landuse = 7]
end

to-report number-school

 report count patches with [landuse = 8]
end

to-report number-traffic

 report count patches with [landuse = 9]
end

to-report number-river

 report count patches with [landuse = 10]
end

to-report number-others

 report count patches with [landuse = 11]
end

