

## Esperimento con Netlogo

di Francesco Galgani

21 luglio 2014

Desidero analizzare, tramite simulazione basata su agenti, un caso di competizione tra individui che può ricordare, seppur nei limiti della modellizzazione proposta, quel che avviene in un sistema economico esclusivamente competitivo e capitalistico, in cui l'unico scopo degli agenti è arricchirsi a spese dell'ambiente e di altri agenti.

E' dato un ambiente composto da  $n$  agenti (*turtles*), ciascuno dei quali ha tre proprietà:

1. “*capital*” = capitale posseduto, inizialmente impostato secondo una distribuzione gaussiana (in modo da simulare una situazione reale) con media pari a 150 e devianza uguale a 50 (*si veda la nota più avanti*);
2. “*chance*” = probabilità di mantenimento del capitale, valore inizialmente casuale da 0 a 100: più è bassa e maggiore è il rischio di una riduzione del capitale, più è alta e maggiore è la possibilità di aumento del capitale;
3. *colore grafico* = rosso se l'agente è molto povero (capitale minore di 10), arancione se il capitale è tra 10 e 99, giallo se il capitale è tra 100 e 199, verde se l'agente è in condizione di benessere (capitale tra 200 e 499), blu se l'agente è molto ricco (capitale maggiore di 500).

L'ambiente ha un vincolo:

- la somma di tutti i capitali posseduti dagli agenti deve essere minore o uguale ad un certo valore “*max*” disponibile nell'ambiente (pari a  $n$  moltiplicato per 1000); ne segue che esiste un valore “*remaining*” corrispondente alla differenza tra “*max*” e la somma dei “*capital*” posseduti dagli agenti.

Ad ogni interazione, ciascun agente si comporta nel modo seguente:

1. sottrae risorse all'ambiente, prendendosi una parte di “*remaining*” (se maggiore di zero) pari alla propria “*chance*”
2. sottrae risorse a ciascun agente confinante (collocato cioè ad una distanza massima di  $k$  caselle) e con “*chance*” minore della propria, prendendosi una parte del suo “*capital*” pari alla differenza tra le “*chance*”;
3. si sposta in maniera casuale di  $k+1$  patches;
4. varia la propria “*chance*” sommandole un numero casuale da -2 a +2.
5. aggiorna il proprio colore in base al capitale posseduto;

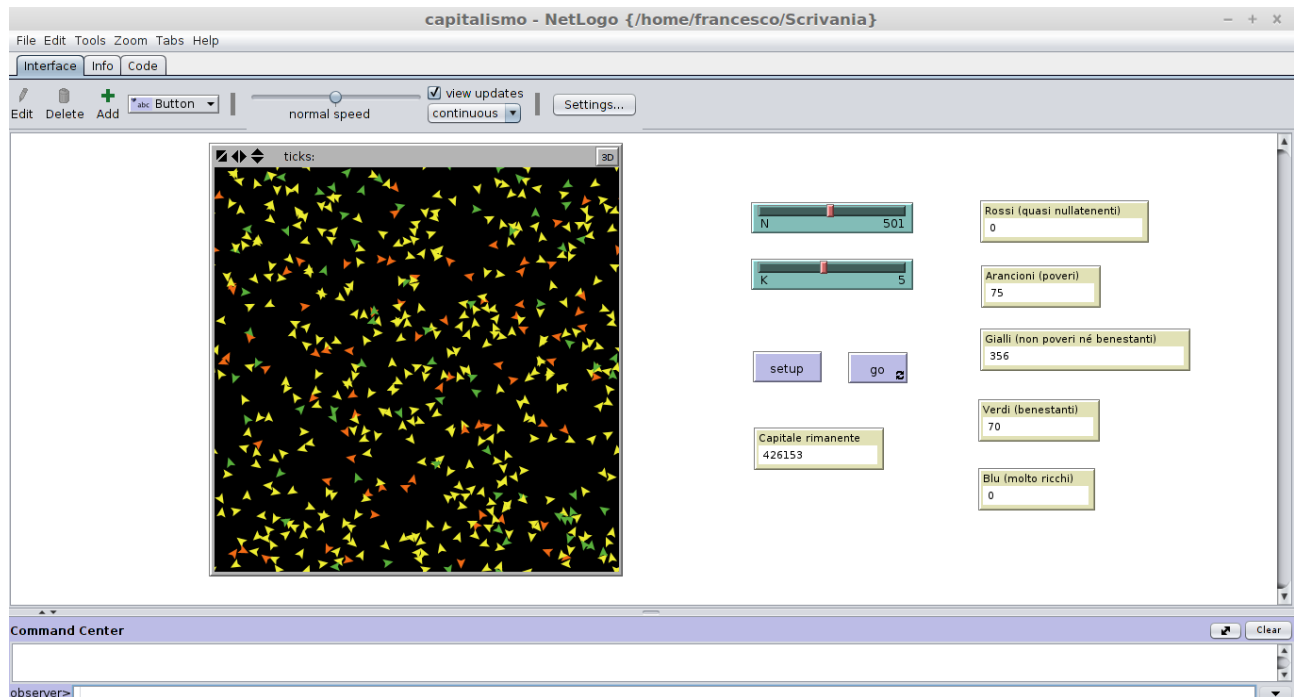
**Considerazioni:** è evidente che le proprietà “*chance*” e “*capital*” rappresentano le disparità sociali, mentre “*max*” rappresenta i limiti ambientali e la finitezza delle risorse: il fatto che “*max*” sia pari a  $n$  moltiplicato per 1000 assicura che ci sia una ricchezza ambientale tale da assicurare a tutti gli agenti ampio benessere. Il fatto che ad ogni interazione ci sia una leggera variazione casuale della *chance* aggiunge maggiore imprevedibilità all'andamento del modello.

**La scelta randomica del capitale iniziale:** il tipo di scelta randomica è decisiva nel determinare la distribuzione iniziale del capitale. La distribuzione gaussiana con media pari a 150 e devianza uguale a 50 è tale da ottenere sempre una prevalenza di gialli (*non poveri né benestanti*), un livello quasi uguale di arancioni (*poveri*) e verdi (*benestanti*), e quasi nessun rosso (*quasi nullatenenti*) e né blu (*molto ricchi*). E' quindi una società con disparità iniziali sicuramente significative, ma non eccessive.

**Domanda sperimentale:** cosa accadrà in questo ambiente simulato? Gli agenti diventeranno tutti ricchi e benestanti? La distribuzione di capitale tenderà ad equilibrarsi tra gli agenti oppure si creerà una condizione di squilibrio sociale maggiore di quella iniziale?

La risposta non è né prevedibile né immaginabile a priori.

## L'interfaccia di Netlogo creata per questo esperimento:



## Il Codice Sorgente scritto sulla base del modello:

```
;; author: Francesco Galgani <francesco@galgani.it>
;; This program is free software: you can redistribute it and/or modify it under the terms of the
GNU General Public License
;; as published by the Free Software Foundation, either version 3 of the License, or (at your
option) any later version.
;; This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without
even the implied warranty
;; of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for
more details.
;; You should have received a copy of the GNU General Public License along with this program.
;; If not, see <http://www.gnu.org/licenses/>.

globals [max_capital remaining rossi arancioni gialli verdi blu incremento]

turtles-own [capital chance]

patches-own []

to go
  ask turtles [
    ;; l'agente sottrae risorse all'ambiente, prendendosi una parte di "remaining" (se maggiore di
zero) pari alla propria "chance"
    if remaining > 0 [
      set capital (capital + chance)
      set remaining (remaining - chance)
    ]

    ;; l'agente sottrae risorse a ciascun agente confinante (collocato cioè ad una distanza massima
di K caselle)
    ;; e con "chance" minore della propria, prendendosi una parte del suo "capital" pari alla
differenza tra le "chance"

    ;; azzero l'incremento da effettuare
    set incremento 0
    ask turtles with [ distance myself <= K ] [
      if chance < [chance] of myself [
        set capital (capital - ([chance] of myself - chance))
        set incremento (incremento + [chance] of myself - chance)
      ]
    ]
  ]
  ;; l'agente incrementa il proprio capitale con quello sottratto agli altri agenti confinanti
  set capital (capital + incremento)
```

```

set incremento 0

;; l'agente si sposta in maniera casuale di k+1 patches;
rt random 360
fd (K + 1)

;; l'agente varia la propria "chance" sommandole un numero casuale da -2 a +2
set chance (chance - 2 + (random 4))
]

ask turtles [
  ;; reimposto il colore di ogni agente in base al nuovo capitale
  setcolor
]

;; aggiorno il conteggio degli agenti in base al colore
conteggio

end

to setup
  ;; pulisco l'ambiente
  clear-all

  ;; imposto le risorse massime disponibili nell'ambiente, cioè la massima somma di tutti i capitali
  set max_capital (N * 1000)
  set remaining (max_capital)

  ;; creo gli agenti
  crt N [
    set capital random-normal 150 50 ;; capitale con distribuzione gaussiana
    set remaining (remaining - capital) ;; imposto la risorsa ambientale rimanente
    set chance random 100 ;; chance casuale da 0 a 100
    setxy random-ycor random-ycor ;; posizione casuale nell'ambiente
    setcolor ;; imposto il colore dell'agente in base al capitale
  ]

  ;; aggiorno il conteggio degli agenti in base al colore
  conteggio
end

to setcolor ;; imposta il colore dell'agente in base al suo capitale
  ifelse capital < 10 [ set color red ]
  [ifelse capital < 100 [ set color orange ]
  [ifelse capital < 200 [ set color yellow ]
  [ifelse capital < 500 [ set color green ]
  [ set color blue ]]]]
end

to conteggio ;; aggiorno il conteggio degli agenti in base al colore
  set rossi count turtles with [color = red]
  set arancioni count turtles with [color = orange]
  set gialli count turtles with [color = yellow]
  set verdi count turtles with [color = green]
  set blu count turtles with [color = blue]
end

```

### Risultato sperimentale:

Il risultato ottenuto in vari esperimenti ha mostrato sempre lo stesso esito, indipendentemente dal valore di  $n$  e di  $k$ , ovvero tutta la ricchezza disponibile tende a distribuirsi in due fasce di popolazione all'incirca uguali rispetto al numero di agenti, ovvero i rossi e i blu, cioè i *quasi nullatenenti* e i *molto ricchi*. Il fatto che le due fasce siano numericamente quasi uguali è probabilmente da attribuire al fatto che le *chance* sono distribuite in maniera puramente pseudo-casuale. ***La società proposta in questo modello crea la massima disparità possibile nella distribuzione delle risorse.***