Simulating the Social Processes of Science workshop
Leiden, Netherlands, April 7th-14th, 2014

Agent-Based Models of Science:
A Glimpse of the Past and Tools for the Future

Nicolas Payette

PhD Student in Philosophy, Université du Québec à Montréal

(Until recently:) Predoctoral Scholar, Center for Connected Learning and Computer-Based Modeling
Plan

I. An Overview of Some Previous Agent-Based Models of Science

II. Modelling Science with NetLogo: Three New Extensions
Part I

An Overview of Some Previous Agent-Based Models of Science
Part I

An Overview of Some Previous Agent-Based Models of Science
Part I

An Overview of Some Previous Agent-Based Models of Science

- Not detailed
- Not many
Part I

An Overview of Some Previous Agent-Based Models of Science

- Not detailed
- Not many
- Not recent
Chapter 4: “Agent-Based Models of Science”
Lotka’s Law

• The number of authors making $n$ contributions is about $1/n^2$ of those making one contribution.
A first, simple model

1. Select a random number from a uniform distribution from 0 to 1. If this number is less than $\alpha$, give the publication to a new (i.e. previously unpublished) author.

2. If the publication is not from a new author, select a paper randomly from those previously published and give the new publication the same author as the one so selected.
A more complex model

• Every paper contains a \textit{kene}: a string of bits, optionally mapped to a 2D space
• At each time step, every existing paper has a small constant probability of reproducing itself
• Author is assigned like in previous model
• Paper has references, chosen at random from “neighboring” kenes, that “pull” it in their direction
Ideas are not created equals

• Differences in “clarity, insightfulness, empirical evidence, theoretical results, application potential.”
  – “Communal functions” vs. “subjective functions”

• Two tasks for the agents:
  – choosing the focal idea
  – choosing the pull ideas.
CLARION Cognitive Architecture
“We put more distance between mechanisms and outcomes, which makes it harder to obtain a match with the human data. Thus, the fact that we were able to match the human data shows the power of our cognitive agent-based approach compared to traditional methods of simulation.” (p. 325)
Distribution of cognitive labor

• Controls
  ...are basically “hill climbers”: they set a direction and move forward as long as they get better results. If they get worse results, they backtrack and change direction.

• Followers
  ...look around them to see if previous agents have found better approaches in their neighborhood and move there if there are. If not, they will look for unvisited place or choose at random.

• Mavericks
  ...first look for unvisited spots. Only if there are none will they move at the best visited place in their neighborhood.
Follower Dynamics

In Complex Adaptive Systems and the Threshold Effect: Views from the Natural and Social Sciences.
Social structure matters

“How does an individual figure out the structure of the world? The truth is that no individual does. It is cultures and communities that plumb the structure of reality; individuals figure out the structure of the world only as they participate in the epistemic networks in which they are embedded.”
Some landscapes are harder than others
Network structure performance
Part II

Modelling Science with NetLogo

Three New Extensions
NetLogo is a multi-agent programmable modeling environment. It is used by tens of thousands of students, teachers and researchers worldwide. It also powers HubNet participatory simulations. It is authored by Uri Wilensky and developed at the CCL. You can download it free of charge.

What can you do with NetLogo? Read more [here](#). Click [here](#) to watch videos.

Join mailing lists [here](#).

NetLogo comes with a large library of sample models. Click on some examples below.
extensions [ nw rnd landscapes ]
patches-own [ value ]

to setup
  clear-all
  setup-landscape
  setup-network
  reset-ticks
end
Global variable: landscape

Choices:

"3 POT HOLES"
"ACKLEY'S FUNCTION"
"ACKLEY'S PATH FUNCTION 10"
"AXIS PARALLEL HYPER-ELLIPSOID FUNCTION"
"BOHACHEVSKY'S FUNCTION"
"BRANIN'S RCOS FUNCTION"
"CPF1"
"CPF2"
"DE JONG F1"
"EASOM'S FUNCTION"
"EUCLIDEAN"
"EXP"
"F3"
"F4 (PHUBERT1)"
"F5 (PHUBERT2)"
"F6 (QUARTIC)"
"F7 (SHUBERT FUNCTION)"
"G3"
"GENERALIZED GRIEWANK FUNCTION"
"GENERALIZED Himmelblau's Function"
"GENERALIZED PENALIZED FUNCTION 1"
"GENERALIZED PENALIZED FUNCTION 2"
"GENERALIZED RASTRIGIN'S FUNCTION"
"GENERALIZED ROSENBRUCK'S FUNCTION"
"GENERALIZED SCHWEFELS PROBLEM 2.26"
"GOLDSTEIN-PRICE'S FUNCTION"
"GRIEWANK'S FUNCTION 8"
"HANSENS FUNCTION"
"HORN'S FMNEASY"
"HORDS 5 PEAKS (MODIFIED)"
"LANGERMAN'S FUNCTION 11 (M=4)"
"LANGERMAN'S FUNCTION 11 (M=7)"
"MS (Himmelblau's Function)"
"M6 (Shekel's Foxholes)"
"MICHALENCOZ'S FUNCTION 17"

example: "a" "b" "c" 3 4 5

OK  Apply  Cancel
(XXV) Generalized Schwefels Problem 2.26
(XXVI) Goldstein-Price's Function
(XXVII) Griewank's Function 8

(XXVIII) Hansen's Function
(XXIX) Horn's FMMEasy
(XXX) Horns 5 Peaks (Modified)

(XXXI) Langermann's Function 11 (M=4)
(XXXII) Langermann's Function 11 (M=7)
(XXXIII) M5 (Himmelblau's Function)

(XXXIV) M6 (Shekel's Foxholes)
(XXXV) Michalewicz's Function 12
(XXXVI) Moved Axis Parallel Hyper-Ellipsoid Function
to setup-landscape
    resize-world 100 100 -100 100
    set-patch-size 3
    landscapes:generate landscape "value"
    ask patches [  
        set pcolor scale-color green value 0 2  
    ]
    ask patches with-max [ value ] [  
        set pcolor red  
    ]
end
Global variable: network-generator

Choices:

"nw:generate-preferential-attachment turtles links population"
"nw:generate-ring turtles links population"
"nw:generate-star turtles links population"
"nw:generate-wheel turtles links population"
"nw:generate-lattice-2d turtles links sqrt population sqrt population false"
"nw:generate-small-world turtles links sqrt population sqrt population 2 false"
"nw:generate-random turtles links population 0.1"

Example: "a" "b" "c" 3 4 5
Preferential attachment
Ring
Wheel
Lattice
Small World
to setup-network
  run network-generator
  ask turtles [ 
    set color yellow
    set shape "person"
    set size 6
    setxy random-pxcor random-pycor
  ]
  ask links [ set color white ]
end
to go
   if-else target != nobody and [ value ] of target > value [ face target
      move-to patch-ahead 1
   ]
   [ move-to one-of patches in-radius 1 with-max [ value ] ]
]
if all? turtles [ value = 1 ] [ stop ]
tick
end
Experiment name: experiment

Vary variables as follows (note brackets and quotation marks):

`["network-generator" "nw:generate-preferential-attachment turtles links population" "nw:generate-ring turtles links population" "nw:generate-landscape" "3 POT HOLES" "ACKLEY'S FUNCTION" "ACKLEY'S PATH FUNCTION 10" "AXIS PARALLEL HYPER-ELLIPSOID FUNCTION" "BOHACHEVSKY'S FUNCTION" "population" 100]

Either list values to use, for example:

`["my-slider" 1 2 3 4 5 6]`

or specify start, increment, and end, for example:

`["my-slider" [0 10]]` (note additional brackets)

to go from 0, 1 at a time, to 10.

You may also vary max-pcor, min-pcor, max-pycor, min-pycor, random-seed.

Repetitions: 100

run each combination this many times

Measure runs using these reporters:

one reporter per line; you may not split a reporter across multiple lines

☑ Measure runs at every step
if unchecked, runs are measured only when they are over

Setup commands:

```
setup
```

Go commands:

```
go
```

Stop condition:
the run stops if this reporter becomes true

Final commands:
run at the end of each run

Time limit: 10000

stop after this many steps (0 = no limit)
Get the model at:
https://github.com/nicolaspayette/sspos