Civil Violence
A Computational Model

Magnus Erik Hvass Pedersen (971055)
Daimi, University of Aarhus, June 2004

1 Introduction

The purpose of this document is to verify attendance of the author to the *Multi-Agent Systems and Co-Evolutionary Games* course at DAIMI, University of Aarhus.

The computational model for civil violence from [Epstein] is first conceptually and mathematically summarized, and then its implementation is detailed. The terminology used here, differs slightly from that of the original.

Unfortunately, the original paper has few concrete results, and is more of an experimental case study on how a few variables and simple interactions between agents, can result in patterns similar to outbursts of civil violence. That there appears to be some statistic regularity, is documented however.

2 Civil Violence

The dynamics of rebellion in a population is the focus of attention. More specifically, we are given a number of agents that are either civilian or cops, and that have simple rules for moving and acting within their habitat.

The civilian agents can go from quiescent citizens to active rebels without any centralized leadership, but solely based on their own (composite) urge to rebel, and the assessment of their local environment in terms of cop- and rebel-densities, since rebelling civilians can be arrested and put in confinement by the cops.

2.1 Mathematical Model

A number of factors combine to give a measure of whether a civilian agent will rebel or not. Although the notion of probability does occur, the model is not probabilistic in the direct stochastic sense of the word. Indirect stochastic rebellion arises from the randomized movement and selection of agents, but for example the probability of arrest for a rebelling agent (as described below), is really used in a threshold-based activation and not as a stochastic tendency.

2.1.1 Hardship

First denote a civilian agent's hardship by H. This is a measure of how toilsome the life of the agent is, the intention being that the more toilsome the higher probability of rebelling. For this model it is a random uniform deviate number:

$$H \sim U(0,1)$$

Each agent has a new H drawn at initialization, and it does not change throughout an experiment.

2.1.2 Legitimacy

Now denote the legitimacy by L. This models how the agents collectively perceive some central authority. Although an agent may experience much hardship, if the legitimacy of the government is high, the individual will not feel that rebellion is justified. For this model, the legitimacy must be within the range:

$$L \in [0, 1]$$

where $L = 0$ means the regime is considered crooked as a bowl of snakes, and $L = 1$ means it enjoys complete faith from its citizens.

2.1.3 Grievance

Hardship and legitimacy may be combined into a civilian agent's grievance $G \in [0, 1]$:

$$G = H(1 - L)$$

where higher G means more grievance. For example setting $L = 1$ we get that $G = 0$, that is, no grievance exists in this model when the goverment is absolutely trustworthy. Conversely when setting $L = 0$, an agent's grievance equals its hardship.

2.1.4 Risk Aversion

Even though a civilian agent may feel grievance, some are more reluctant to burst into rebellion than others. This is modelled by R, the agent's level of risk aversion. As with the hardship H, it is also a random number drawn for each civilian agent:

$$R \sim U(0, 1)$$

Again, this is kept throughout the entire experimental run.

2.1.5 Cop-To-Active Ratio

The risk an agent is willing to take, should be held against the possibility that a nearby policing agent - a cop - may arrest the agent if it rebels. That is, let $(C/A)_v$ denote the cop-to-active ratio, so that within the agent's neighbourhood or vision v, the number of cops is divided with the number of civilians that are (actively) rebelling.

2.1.6 Arrest Probability

An agent considers the behaviour of its neighbourhood through the cop-to-active ratio $(C/A)_v$ before turning rebellious, the rationale being that the more civilian agents that are rebelling and the less cops there are to arrest them, the

less likely it is that the agent will be arrested if it goes active and participates in the rebellion. The probability P of the agent being arrested, is taken to be:

$$P = 1 - exp[-k(C/A)_v] \tag{1}$$

With the constant $k \in \mathbb{R}$ chosen so that P is plausible when $(C/A)_v = 1$. For these experiments $k = 2.3$, meaning that $P \simeq 0.9$ in case of an equal number of cops and active rebels.

2.1.7 Net Risk

Finally, the agent's reluctance to take risk, and the probability of it being arrested if it goes active, is combined into the agent's net risk N:

$$N = RP$$

Which is directly used in the behavioural rule for a civilian agent in section 2.2.

2.2 Rules

Let M denote the rule of movement which is common to both civilians and cops, and in which the agent moves to a free random position in its vision v.

For a civilian agent to become an active rebel, we have the rule A, which is given by:

$$A : \text{if } (G - N > T) \text{ rebel; otherwise be quiet.}$$

where $T \geq 0$ is a non-negative activation-threshold.

Finally, the cop-rule denoted by C, is to have a cop randomly arrest a rebel in its vision v, and put it into confinement and hence removing it from the game for a certain duration.

3 Behavioural Details

Even though the rules $\{A, C, M\}$ dictate the behaviour of the agents, there are still some details that need to be adressed.

All agents are initially placed randomly in a rectangular grid called the arena, with each agent occupying one cell. Assuming an agent is placed in the cell at position (x, y), the cell-positions within its vision v, are the following:

$$\{(x + i, y + j) | i, j \in \{-v, \cdots, v\}\}$$

within the boundaries of the grid of course.

Each step in updating the state of the model consists of randomly selecting one agent in the arena, and after applying the movement rule M, then either use rule C if the agent is a cop, or use rule A if the agent is a civilian.

3.1 Rebel Activation

It is important to stress that the rebel-activation rule A, is only employed after movement of the civilian agent in question. This means that a group of civilian agents could become active rebels, and then by coincidence all but one could move away from that region. Even though the civilian agent that is left may have low grievance and primarily rebelled because its neighbours did, it remains a rebel since it does not employ rule A until it actually moves.

The reason for only using rule A after movement of an agent, is that it is self-referential in its use of P from Eq.(1). More specifically, since P counts the number of active rebels in the agent's neighbourhood, and this is used in deciding whether the agent should also rebel, and this in turn is used to decide whether the neighbouring civilian agents should rebel - and so on and so forth - one must decide in which order to evaluate the agents.

The natural choice seems to have agents update their status after movement - that way there is no choice of the order of evaluating agents' P in the grid, be it from left-to-right, up-to-down, or vice versa. Because of this, all civilian agents are also initially quiescent.

3.2 Jail

When a cop arrests an active rebel and puts it in confinement, a random jail-term up to J_{max} is selected. The duration is measured in agent-steps, as described in section 3. That is, before the random selection of an agent for movement, the jail is processed. This is done by decreasing the jail-term for each imprisoned civilian agent, and upon reaching zero, releasing the agent and placing it in a randomly chosen cell in the arena.

At the point of release, the civilian agent uses the activation rule A to decide whether to instantly rebel again, or go quiescent. This model does not alter the grievance of an agent having just been imprisoned, and moreover does not take the fear of imprisonment into account in the rebel-activation rule.

4 Inter-Group Violence

An extension to the model is briefly described in [Epstein] also, in which there now are two groups of civilians, who may kill each other when becoming active rebels. The civilian agents may also reproduce, with the offspring inheriting the parent's ethnic identity and grievance. Since there is growth in the number of agents, eventually they must also die of old age to avoid overpopulation.

It could be interesting to model Darwin's theory of *Survival of The Fittest* in this context, thus resembling *Genetic Algorithms*.

When legitimacy decreases, the groups start rebelling and one group may end up wiping out the other - which is in fact always observed but with random victor. The presence of a police force may prevent such genocide from occuring, but more interesting perhaps, is that there may emerge patterns akin to safe

havens, where one ethnic group is protected from the other by the policing agents.

This extension to the model is not described in enough detail for implementation, but its analysis is still of some interest.

4.1 Dynamic Analysis

In the vein of [Hofbauer & Sigmund] one could try and analyse this rivalry. Although the present model is significantly more complex, a simplification that perhaps could yield some insight, would be to study the density of civilian agents in each group, denoting them x and y and each belonging to the range $[0, 1]$. Using population density instead of population size, implicitly takes the size of the arena into account, for example $x = 1$ would mean the entire arena is populated by the first group of civilians. A third density measure for the cops could be $z \in [0, 1]$.

Again, the exact locations of agents are overlooked, and their mere densities in the arena the basis of analysis. It could furthermore be advantageous to introduce $x_r \leq x$ and $y_r \leq y$ which are the densities of the rebelling parts of the two groups x and y. A simple linear relationship between the current number of agents x and the number of agents in the next step x', is then given by the birth of new individuals, the death of agents killed by rebelling rivals, killed or arrested by police, and the death that is due to old age:

$$x' = x + ax - bxy_r - cxz - dx$$

with $a, b, c, d \geq 0$. Which may be simplified to the following:

$$x' = x(1 + a - by_r - cz - d)$$

The dynamics of rebellion are then modelled in a separate time-series, and could for example take into account the population density as well as the previous rebel-rate, and some average of the net-risk $\overline{N}_x$ for that population-group, as well as the density of cops z:

$$x'_r = ex + fx_r + g\overline{N}_x - hz$$

again with $e, f, g, h \geq 0$.

It is of course beyond the scope of this document, but a rigorous analysis of such a simplified version of the computational model, could result in categorization of the parameters $a, \cdots, h$ as well as the legitimacy L and so forth, in terms of the effect they have on the population densities (x, y). That is, when do the two densities oscillate, when does one dominate the other, when are they chaotic, and is it actually possible within some range of parameter choice, to find a socalled rest point $\mathbf{F} = (\overline{x}, \overline{y})$, where the population densities would remain $\overline{x}$ and $\overline{y}$ for each step, and possibly even be a point of attraction or convergence, so that stable coexistence is always achieved within some ranges of the parameters and initial choices of densities $x[0]$ and $y[0]$.

5 Implementation

The computational model is implemented using *Microsoft Visual C++ .NET* and compiles into an executable for *MS Windows*. Source-code and executable (`CivilGUI.exe`) can be found at `http://www.daimi.au.dk/~u971055/`.

5.1 Arena & Agent Classes

The arena is implemented in `LArena` and the base-class for the agent is named `LAgent`. The sub-class for the cop is `LCop` and specializes the agent-behaviour correspondingly, and similarly the sub-class for the civilian agent is named `LCivilian`.

The cells of the arena are implemented as one single array. An agent's placement in the arena is then uniquely identified by the cell's index into this array.

5.2 Graphical Display

In the original paper, two screens are used for displaying the agents' type and state. Here however, only one screen is used, and the state of the game is visually presented as follows.

The arena is depicted as a white rectangle and the agents are then small coloured boxes, with cops being solid blue whereas civilian agents have two of their variable features displayed: The grievance which goes from green to red[1] depending on its degree, and this is surrounded by a frame coloured by the agent's state of rebellion; red if rebelling, green if quiescent.

5.3 Parameters & Buttons

The following parameters may be altered while running the program. Step-size, or the number of agent-steps that are executed upon pressing the "Step"-button. The rebel-activation threshold T, the legitimacy L which is common to all agents, the constant K used in calculating a civilian agent's arrest probability. The vision v which is common to both cops and civilians. The maximum prison sentence J_{max}, and finally the threshold of revolution which is further described in section 5.4.

It is possible to alter these parameters during the course of an experiment. For example, the legitimacy can be changed and its effect on grievance can be seen right away without performing any agent-steps. As civilian agents only have their rebellion-status updated after movement however, it will not be immediately affected by the change in grievance.

By pressing the "Reset'-button, the arena is replaced by a newly created one, thus restarting the experimental run. This is useful in conducting several experiments for the statistics in section 5.4, that can be dumped to a file named

[1] The intensity of red being the squareroot of G, and the intensity of green being the squareroot of $1 - G$.

`CivilGUI.txt` by pressing the "Output"-button. The whole process of running a number of experiments and dumping the statistical output, can be executed by entering the number of runs and pressing the "Run"-button.

There is little or no error-correction in the input, and the user must ensure that input values are within their correct ranges.

5.4 Statistical Output

That some statistical regularity exists in the model, can be illustrated by counting the number of steps between outbursts of revolution, that is, when the number of active rebels is beyond some threshold. This bookkeeping is done by the `LStatisticsThreshold` class.

5.5 Data Structures

For larger arenas, efficient data structures are necessary, since a naive implementation would require computational time proportional to the number of cells in the grid for many of the algorithms.

For example in the random selection of a free cell upon the release of a civilian agent, a direct approach would each time gather all of the arena's free cells in a separate array, and then select one at random. Instead however, such a separate list of the free cells is kept and maintained, with the selection of a random element then taking constant time $O(1)$, because the arena should not be traversed first to create the array. When agents move, the array is simply updated appropriately, and also with constant time usage.

Since only one agent is moved per step, a list of live agents is also maintained, again with the intention of quickly being able to select one at random. Civilian agents that are imprisoned are removed from this list and inserted again upon their release from prison. These operations also take constant time.

Some bookkeeping is necessary for these tricks to work, but is deemed worthwhile - even more so if the arena-size and number of agents were to grow. The result is that the model executes thousands of steps in the experiments below, with only marginable delay in program response.

5.6 Arena Traverser

Although $O(1)$ algorithms are used in many cases, it is more practical to use actual traversal when doing calculations involving an agent's neighbourhood, as is the case with the selection of a free neighbouring cell, a cop's selection of a neighbouring rebel, and the calculation of the cop-to-active ratio.

The approach taken here is object-oriented, so that the base-class `LArenaTraverser` implements the basic traversal, which is then specialized in a number of subclasses. For example the sub-class `LSelectionTraverser` which implements the random selection mechanism used in `LRebelTraverser` for finding a neighbouring rebel, and `LFreeCellTraverser` for finding a free neighbouring cell. The

base-class is also inherited in `LCopToRebelRatio` which calculates the cop-to-active ratio.

The running time of the basic traversal algorithm and hence all of its descendants, is $O(v^2)$ with v being the agent's vision. This could be lowered by clever usage of data-structures, but the increased complexity of implementation and maintenance, is not justified by the modest decrease in running-time for the small v used here.

5.7 MS Windows Details

The entry point for the Windows application is the `_tWinMain()` function found in `CivilGUI.cpp`, that sets up the graphical user interface (GUI) in the default manner, with buttons and text-input fields being added manually in the source-code.

The drawing of the arena with its agents, and the displaying of status-text is done in the `DoPaint()` function.

The pressing of a button or change of text, sends a corresponding message to the `WndProc()` function, which calls the appropriate functions. The function `DoGetWindowTextI()` is used in retrieving the multitude of integer-valued parameters, and `DoGetWindowTextF()` for the real-valued parameters. The functions `DoPaint()`, `DoStep()`, and `DoReset()` are called for updating the graphical display, performing an agent-step in the model, and resetting the model, respectively. The function `DoRun()` iteratively calls `DoReset()` and `DoStep()` to perform the given number of experimental runs.

Since the program is adequately fast for experimental use, there is no multi-threading implementation of the simulation-part and the GUI, which would be necessary if the GUI should be accessible while running a series of consecutive agent-steps in the simulation.

5.8 Random Number Generator

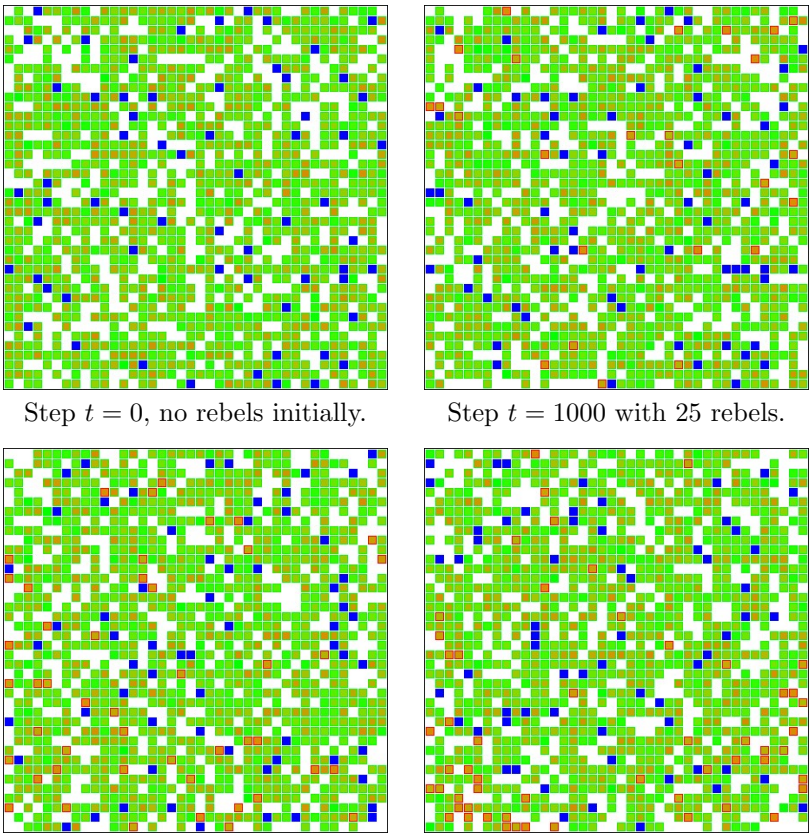
Random numbers are generated using the `ran1` procedure from [Press, et al]. Its initial seed is taken from the `clock()` system-call, which returns a number relative to when the system was booted. In short, it will generally be different for each run of the program, so reproducible experiments will require a recompilation with a hardcoded seed.

6 Experimental Results

The default parameters are as follows: The legitimacy is $L = 0.3$, the rebel activation threshold $T = 0.45$, the constant $K = 2.3$, the maximum prison sentence is $J_{max} = 30$ agent-steps, and an agent's vision in each direction is $v = 7$. The arena consists of 40×40 cells, with a civilian population density of 0.7, and 0.04 for the cops.

6.1 Example Run

Using these default parameters, the state of the arena is displayed in figure 1 for various time-steps. Because all civilian agents are initially quiescent, it takes some time for the number of rebels to rise to the point of revolution.



Step $t = 0$, no rebels initially. Step $t = 1000$ with 25 rebels.

Step $t = 5000$ with 47 rebels. Step $t = 50000$ with 55 rebels.

Figure 1: Snapshots of the arena for various time-steps.

6.2 Statistical Regularity

A revolution is said to occur when there are more than 50 civilian agents that are rebelling. The result of counting the frequencies of different durations between outbursts of revolution in 50 runs of a million agent-steps each, is presented in figure 2. Because the maximum waiting-duration is over 60000 agent-steps, the graphs below are truncated to depict only up until 100 agent-steps, beyond which the frequencies are too low to show up with a linearly scaled y-axis.

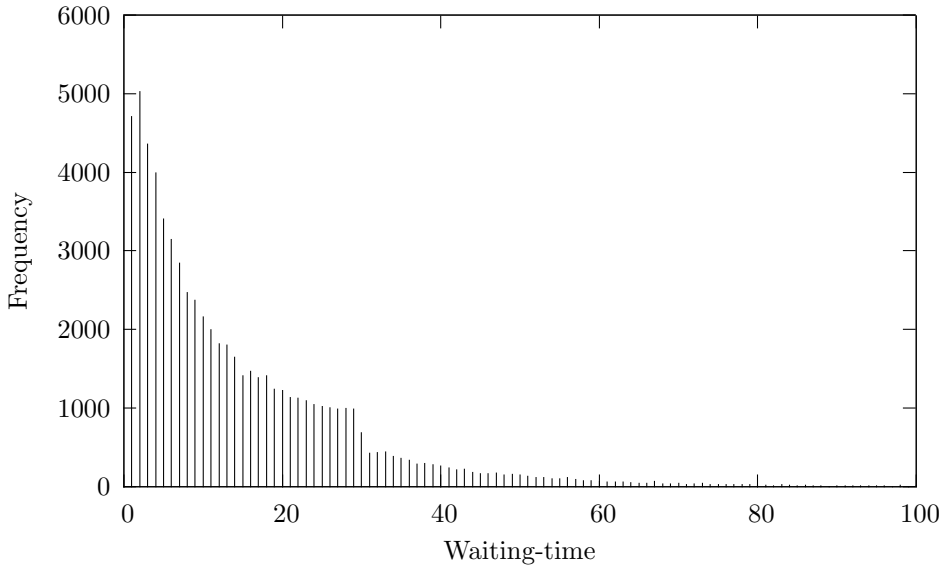


Figure 2: Distribution of the waiting time between outbursts of revolution.

Using a logarithmic scale for the frequency- or y-axis, clearly shows that the distribution of waiting times between revolutions is exponentially decreasing, as can be seen in figure 3. Although the data beyond the 100th bin does show up with logarithmic scale, it is erratic and cloudens the fact that the bulk of the distribution is fairly ordered as illustrated by the fitting.

Similarly to the analysis described in section 4.1 for two warring population groups, one could also analyse the dynamics of rebellion for just a single population-group in a simplified density based model. And indeed, one could find that complex oscillatory patterns occur in the revolutionary dynamics, for this and other choices of parameters.

7 Conclusion

The efficient implementation of a computational model for decentralized, agent-based modelling of civil violence was given. Experiments were conducted with the same findings as in the original paper [Epstein], namely that there is some statistical regularity in the timing of the occurence of revolutions amongst the civilian agents.

Ideas for analysing a simplification of the model were also discussed, in which the evolution of the densities for the civilian agents and their rebelling cliques are considered. Suggestions for recurrence relations were given for the variant of the computational model which has two warring populations.

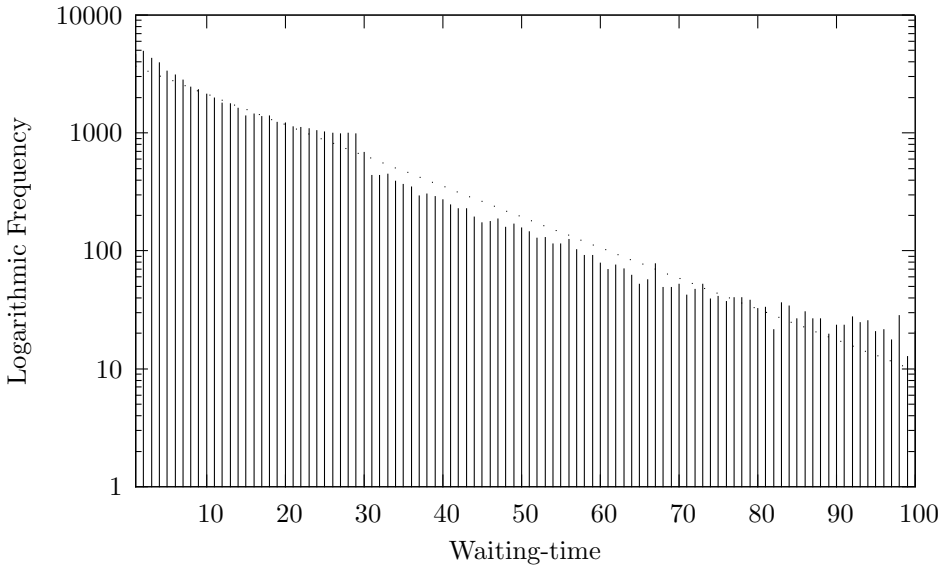


Figure 3: Distribution of the waiting time between outbursts of revolution, but with logarithmic y-axis.

References

[Epstein] Modeling civil violence:
An agent-based computational approach
Joshua M. Epstein
PNAS, May 2002, vol. 99
`http://www.pnas.org/cgi/content/abstract/99/suppl_3/7243`

[Hofbauer & Sigmund] Evolutionary Games and Population Dynamics
Josef Hofbauer, Karl Sigmund
Cambridge University Press 1998
ISBN 0-521-62570-X

[Press, et al] Numerical Recipes in C
Press, Teukolsky, Vetterling, Flannery
Cambridge University Press 1992
ISBN 0-521-43108-5
`http://www.nr.com/`