Interbank relationships after a shock

Hector Castillo hector@bith.net

Table of contents

In the model reflected in `interbank.py` we have built a model that simulates credit relationships between financial institutions, with preferential agreements to have a potential credit channel, used when D of the institution could not be covered by its own assets.

Now, we will introduce restrictions in the credit relationships, to understand what happens after a shock that halts the bank system, and to use it to experiment how to learn a variable that directs the creation of links between banks directed by the regulator.

Previous simulation

Starting links

1. Each banks choose randomly a lender. Only restriction is that lender should not be itself.
2. Figure 1 represents one of those simulations at instant $t = 0$. By design $\hat{d} = 1$, which means we limit to one outgoing link for each bank (not limiting the incoming links). Thus borrowers can only get loans from $\hat{d}$, lenders.

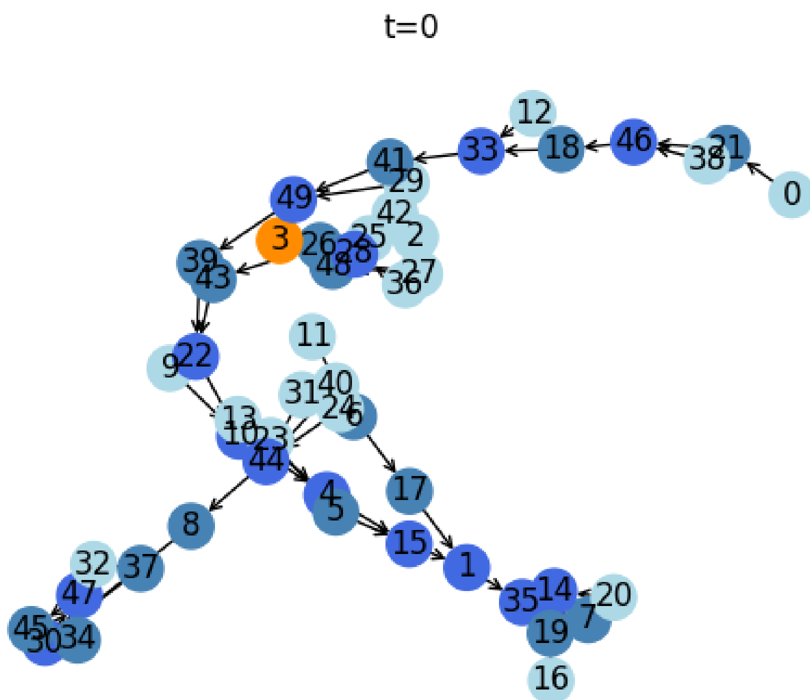


Figure 1: Random initial graph of banks' links

3. In the next instant $t = 1$ the link is changed or not depending on the fitness μ of the current lender and a possible new one (selected again randomly) as Equation 1 indicates:

$$P_t^j = \frac{1}{1 + e^{-\beta}(\mu_t^k - \mu_t^i)} \tag{1}$$

4. Till now what we have is the normal evolution of the model. Let us now introduce the shock.

New algorithm

Now it is separated from the code that evolves the model ($t = 0$ till $t = T$) from a new code that chooses the new lender or allows not to have a lender, named `LenderChange`.

This class `LenderChange` will be responsible of select a method to obtain a new lender for each bank, or neither in some cases, running the model with different results:

- `Boltzman`
- `InitialStability`
- `ShockedMarket`

The first derived class `Boltzman` with the "normal" method described: randomly chosing a new lender using the Boltzman equation described in Equation 1.

`InitialStability`

A variation in which we use a Barabási–Albert model to generate the relationships between banks with initial degree $m_0 = 1$. We have altered the algorithm to obtain a directed graph where the leaves are unidirectionally connected to their precedessors till we arrive to the guru (the node with the most number of incoming connections), as Figure 2 shows.

In this method, the evolution of the model is the same as in `Boltzman`: in each new step, we randomly use a probability p to change the lender, but the preferential to continue with the previous attached lender has been increased to 80%, instead the 50% of the `Boltzman` method. Here all the banks have one and only one outgoing link, so all of them have a possible lender (to obtain money or not will depend on the capacity of that bank).

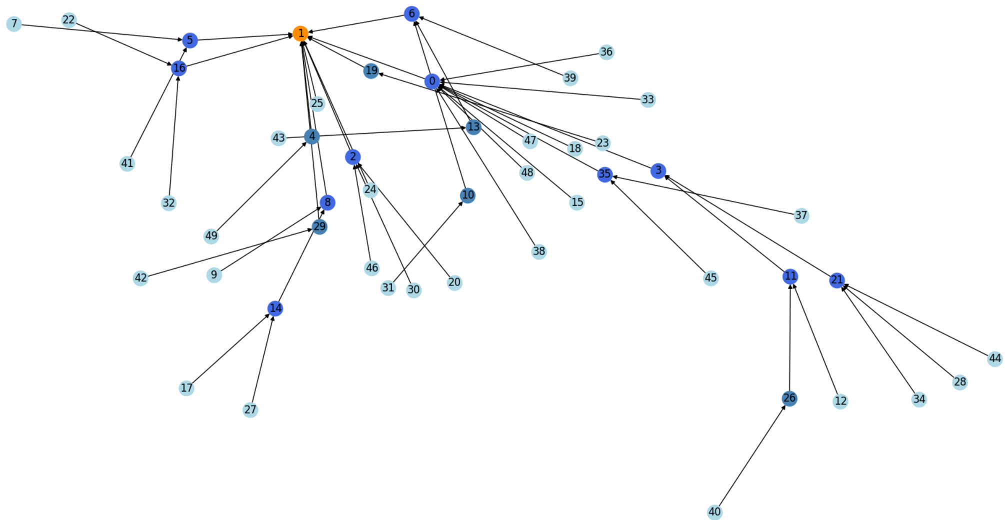


Figure 2: Example of a directed Barabási–Albert graph. Denote the use of orange for the guru and different tones of blue for the rest of nodes depending on the number of incoming connections, till lightblue for the leaves.

`ShockedMarket`

It is a method though to represent a shocked market in which there are few links between banks: no one trust in another, so the majority of them are isolated and no outgoing link. It is created using a Erdős–Rényi like the example in Figure 3.

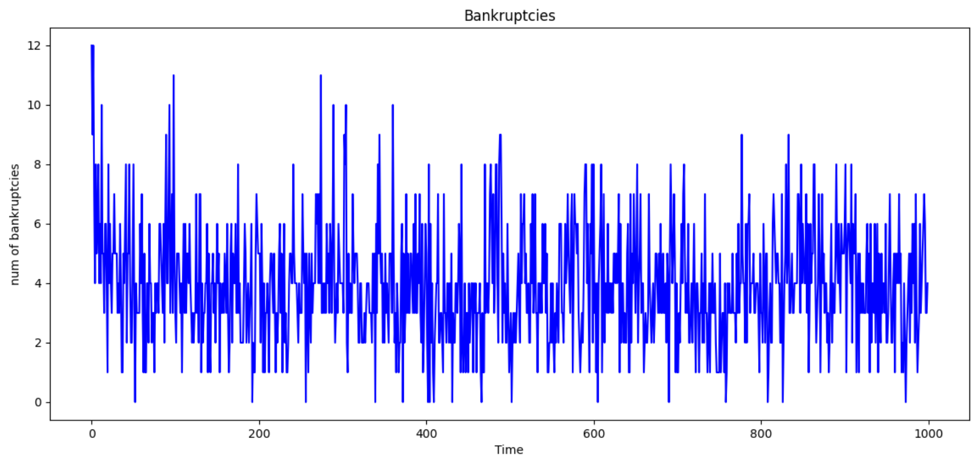
Changes made in the algorithm:

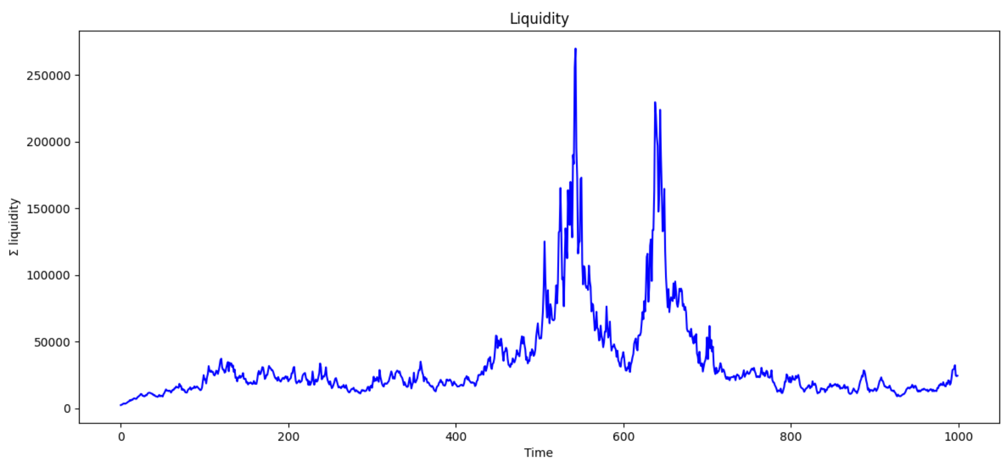
- If initially the bank has no lender, it will never have one: the $t = 0$ situation of lenders are sustained till the end $t = T$.
- Without a lender the balance in this case is harder: deposit shocks conduct faster to bankruptcy.
- If the bank goes bankruptcy, it is created again with the same lender (if it has one).
- Greater the p parameter of the Erdős–Rényi graph generator, more links we will have.

Figure 3: Example of an Erdős–Rényi directed graph with $p = 0.001$. Only five connections between nine banks isolated in three islands .

Results

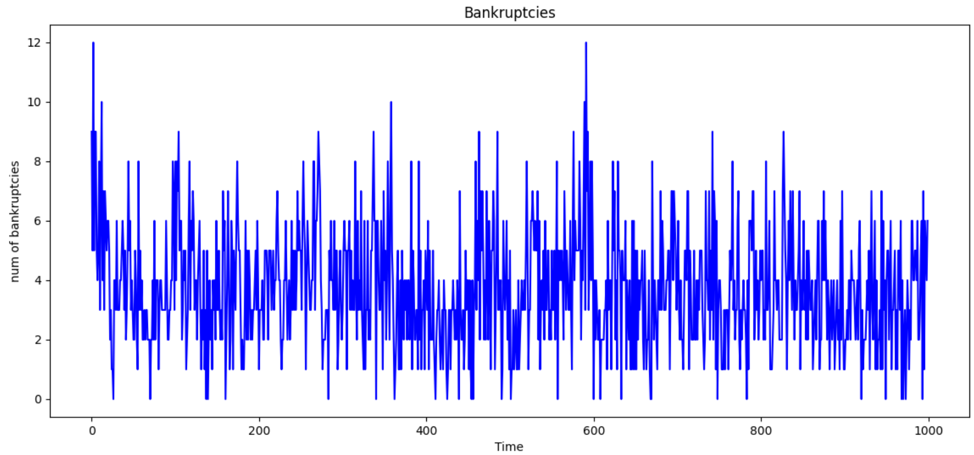
Boltzman

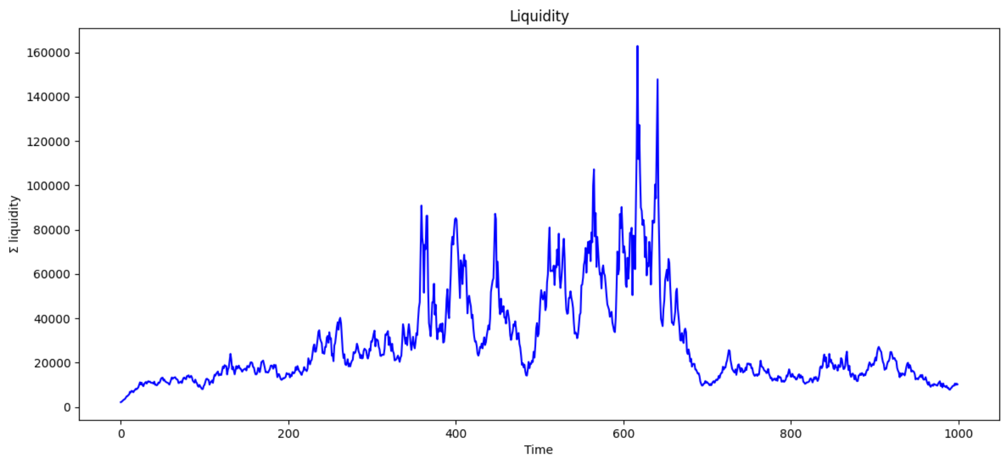




Barabasi

No impact or better perfomance with this idea, so I discarded. The only possibility I think I can try is to increment m_0 to values greater than 1 and obtain nodes with degrees of 2..n links, allowing the change of lender only on those paths.





`ShockedMarket` with $p = 0.001$

When we use $p = 0.001$ or lower values the expected result was a lower perfomance than what here we refer as `Boltzman` method. Using Figure 4 for connect initially the banks, we have only one bank (33) with a lender (19). Obviously the lender is the guru.

But with this situation we have a greater perfomance in terms of liquidity.

`ShockedMarket` with $p = 0.002$

When we use $p = 0.002$ we will obtain more links... and so on.

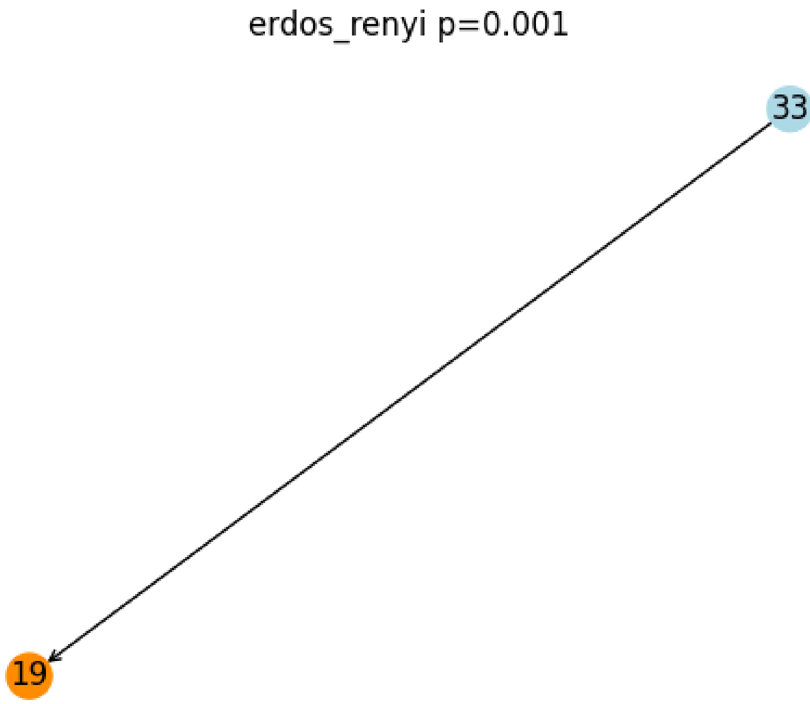


Figure 4: Erdős–Rényi used in this simulation.

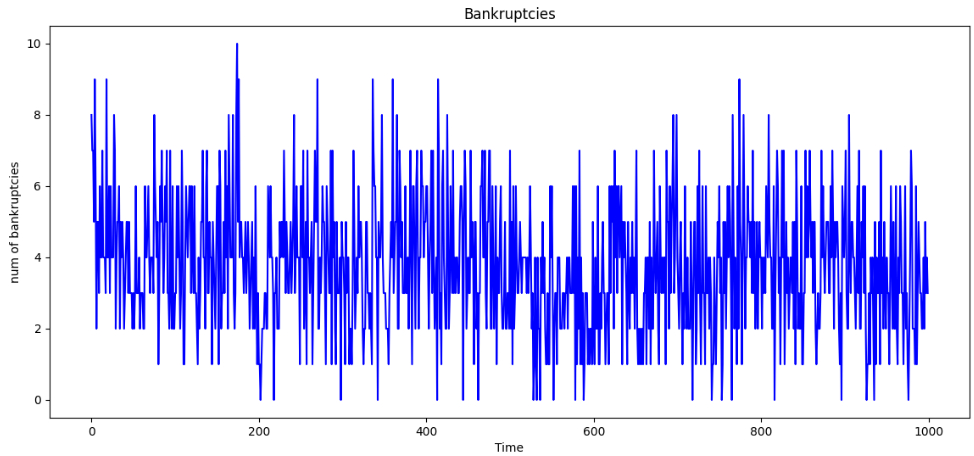


Figure 5: Bankruptcies with $p = 0.001$, with only one link between two banks and the other 48 restricted to operate without loans.

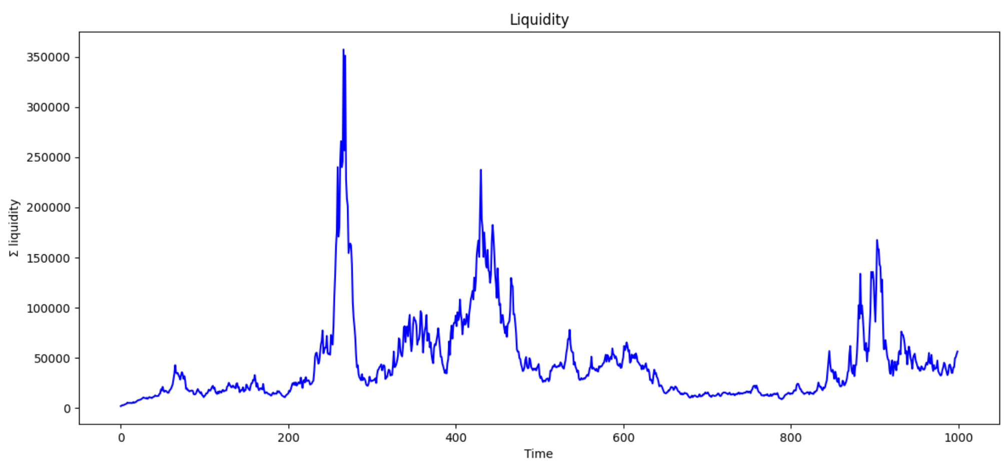


Figure 6: Liquidity with $p = 0.001$

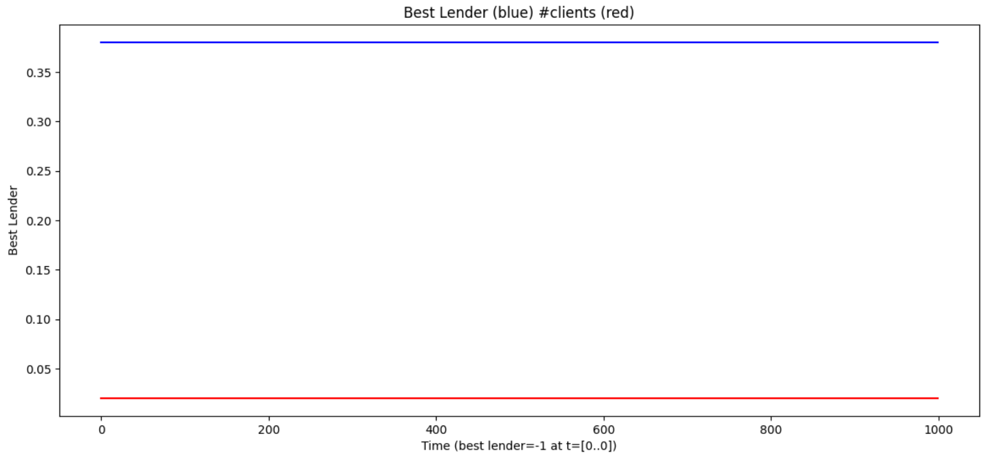


Figure 7: Best lender evolution $p = 0.001$

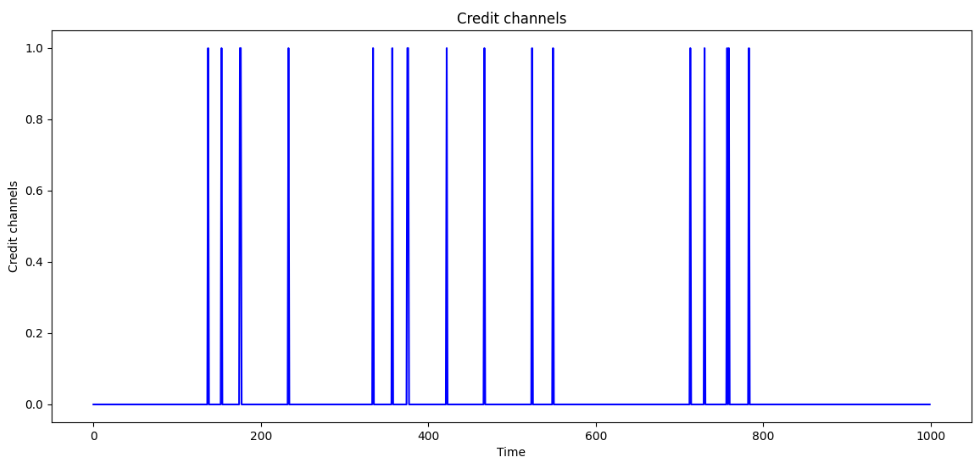


Figure 8: Credit channels $p = 0.001$

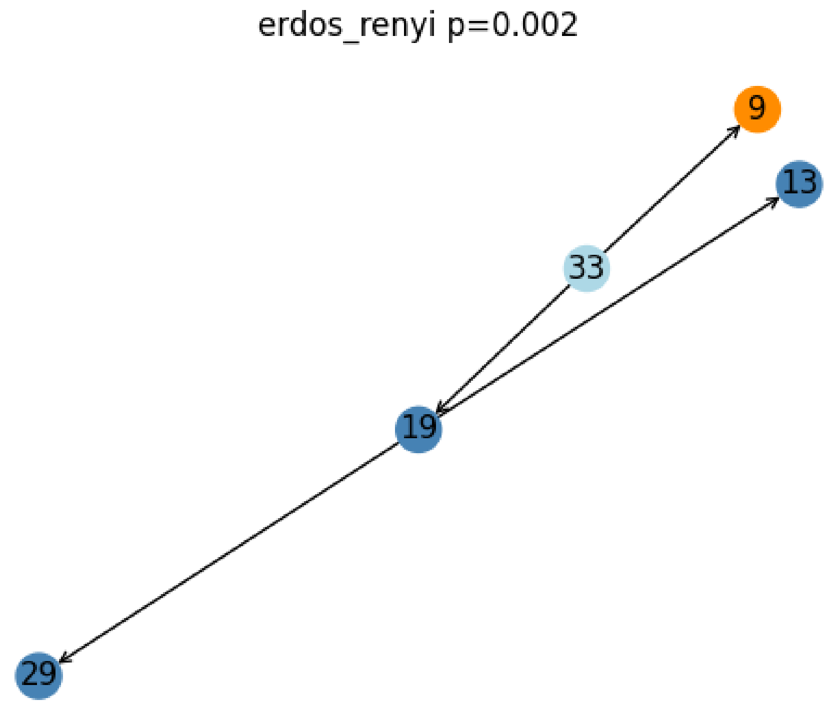


Figure 9: Erdős–Rényi used in this simulation.

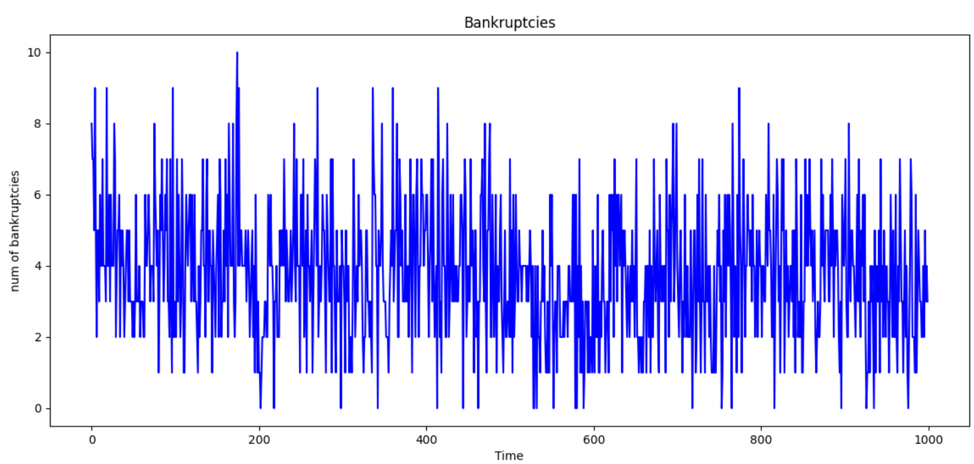


Figure 10: Bankruptcies with $p = 0.002$

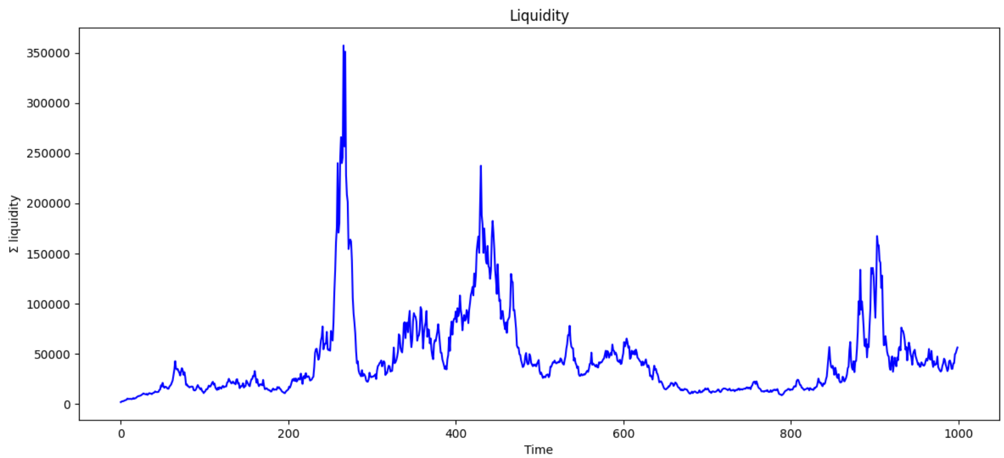


Figure 11: Liquidity with $p = 0.002$

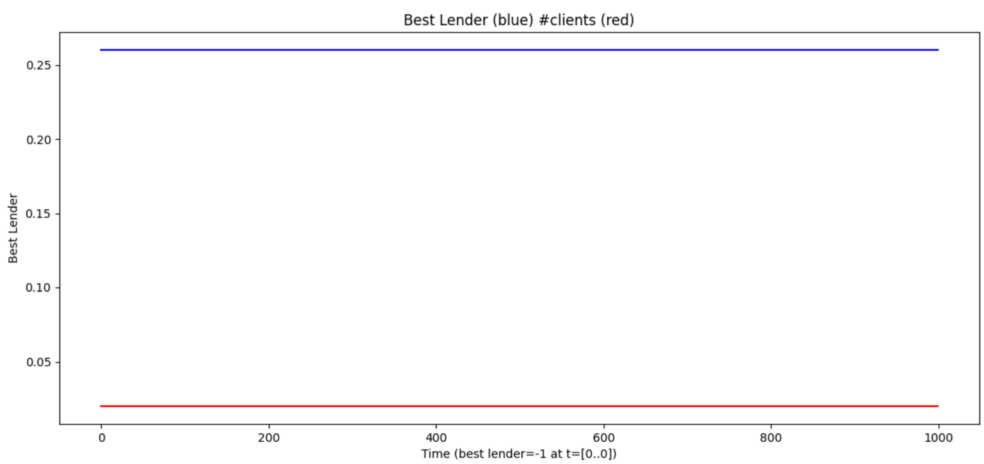


Figure 12: Best lender evolution $p = 0.002$

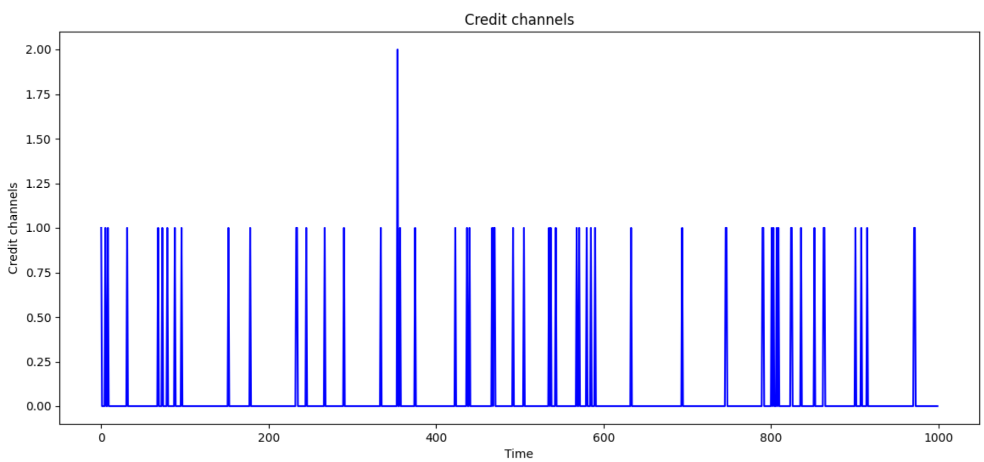


Figure 13: Credit channels $p = 0.002$