Using Cellular Automata on a Graph to Model the Exchanges of Cash and Goods

Ranaivo Mahaleo Razakanirina and Bastien Chopard

University of Geneva, Switzerland
ranaivo.razakanirina@unige.ch,
bastien.chopard@cui.unige.ch

Abstract. This paper investigates the behaviors and the properties of a "Give and Take" cellular automaton on a graph. Using an economical metaphor, this model implements the exchange of cash against goods, among the nodes of a graph G, with a local pricing mechanism. During the time evolution of this model, the strongly connected components (SCC) emerge, mimicking the creation of independent sub-markets. In the steady state, each SCC is characterized by a unique price obeying the supply and demand law for that sub-market. We also show that the distributions of cash and goods are proportional to the indegree of the cells, reproducing a Zipf's law of wealth distribution in case of a scale-free graph topology.

Keywords: Complex system, cellular automata on a graph, complex network, strongly connected components, economical model.

1 Introduction

A complex system is an organization which consists of many parts and the interaction between them [11,3]. This approach gains in popularity and offers a new way to the scientists and the researchers to model complex phenomena in various real world applications. Complex phenomena can hardly be solved analytically by mathematical models and numerical simulations are needed. Complex networks [6] are now widely used to described interaction patterns in social or economical systems. We can cite among other applications the evolution of the structure of complex networks [4] or the modeling the propagation of economic crises [14].

Cellular automata [7] are an effective tool to study complex systems and it is natural to consider their extension to a graph topology. In [8,13] we have defined cellular Automata on Graph (CAG), a formalism that extends the power of classic cellular automata approach by introducing irregularity and dynamics on the neighborhood relationship.

To illustrate the CAG approach, we consider here a particular case of a CAG which we called the "Give and Take" model (GT model), which implements a simple economical interaction between agents. We show that this model, interpreted as a market dynamics, produces interesting results: spontaneous

S. Bandini et al. (Eds.): ACRI 2010, LNCS 6350, pp. 163–172, 2010.

sub-market creation, emergence of a global price in each sub-market, compliance with the supply-demand law and Zipf's type of wealth distribution.

The paper is organized as follows: we first define formally the GT-model, then we discuss its computer implementation on the CAG engine. Finally, we present the simulation results and formulate mathematically the properties of the model.

2 GT (Give and Take) Model Formalism

The GT-model simulates the complex interactions between agents exchanging goods against cash. The model is built with a directed graph G which has n cells and m edges. One cell corresponds to one agent. A directed edge (i, j) of the graph G (see Fig. 1) models the buying and selling actions between i and j: agent i gives cash to buy goods from agent j and, in exchange, j returns a certain quantity of goods.

Cells (or agents) and can have both the role of a buyer or a seller. As a buyer, a cell distributes its cash to each seller it is connected to, in a way which is inversely proportional to the price offered by the seller. In turns, each seller distributes its good in proportion to the money received from each buyer. For each buyer-seller relation, a new price is computed, as the amount of money spent over the amount good received.

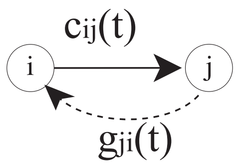

Fig. 1. The flows of cash used to buy goods are indicated by the direction of the edges of the graph G. Here, the flow comes from i to j. The dotted edge represents the direction of the flow of goods in exchange, which is the opposite direction of the flow of cash.

Table 1 summarizes all the notations used to describe the GT-model.

The state of each cell i at iteration t consists of the cash amount $c_i(t)$ and the quantity of goods $g_i(t)$ owned by the cell. We assume that these quantities are infinitely divisible. To prevent the rainy days, at each iteration, each cell invests only a fraction λ_i of its cash and a fraction μ_i of its goods. The amount of cash offered from agent i to agent j is denoted by $c_{ij}(t)$ and the amount of goods given in exchange by g_{ji}. The price proposed by seller j to buyer i is denoted as $p_{ij}(t)$.

The dynamic of GT-model at each time step t is composed of four phases, treated successively as follow:

Table 1. Summary of notation

Symbol	Meaning
$c_i(t)$	Cash owned by cell i at the time iteration t
$g_i(t)$	The quantity of goods owned by the cell i at the time iteration t
λ_i	The fraction of cash invested by the cell i to others at each iteration. $0 \leq \lambda_i \leq 1$
μ_i	The fraction of goods invested by the cell i to others at each iteration. $0 \leq \mu_i \leq 1$
c_{tot}	The total cash in the whole CAG
g_{tot}	The total amount of goods in the whole CAG
$c_{ij}(t)$	The flow of cash from i to j at the time iteration t
$g_{ji}(t)$	The flow of goods from j to i at the time iteration t
$p_{ij}(t)$	The unit price of goods proposed by j at the time iteration t
k_i^{in}	The indegree of the cell i
k_i^{out}	The outdegree of the cell i
N_i^{in}	The set of the incoming neighbors of i or the buyers connected with i
N_i^{out}	The set of the outgoing neighbors of i or the sellers connected with i

– *Giving phase:* During this phase, each cell gives cash to its connected sellers. The buyers obey the following strategy: "give more cash to the sellers that propose a better price". This cash value is inversely proportional to the unit price of goods proposed by the sellers. Mathematically, the resulting cash flow can be expressed as [13]

$$c_{ij}(t) = \frac{p_{ij}^{-1}}{\sum_{l \in N_i^{out}} p_{il}^{-1}(t)} \lambda_i c_i(t), \quad j \in N_i^{out} \ . \tag{1}$$

At $t = 0$, the initial price can be defined randomly, or assumed to be equal for each seller.

– *Taking phase*: In exchange, each cell returns a certain quantity of goods to its buyers. The highest bidder wins the highest amount of goods. This quantity of goods is proportional to the cash received at the giving phase. Thus the the flow of goods is given by [13],

$$g_{ji}(t) = \frac{c_{ij}(t)}{\sum_{l \in N_j^{in}} c_{lj}(t)} \mu_j g_j(t) \quad i \in N_j^{in} \ . \tag{2}$$

– *Self-adapting price*: The unit price of goods at the next iteration $t+1$ is the ratio between the cash given at the giving phase and the quantity of goods in exchange at the taking phase. From (2) we have

$$p_{ij}(t+1) = \frac{c_{ij}(t)}{g_{ji}(t)} = \frac{\sum_{l \in N_j^{in}} c_{lj}(t)}{\mu_j g_j(t)} \equiv p_j(t+1) \ . \tag{3}$$

We see in (3) that the unit price of goods actually depends only on the seller j. In other words, the seller proposes the same unit price of goods to all its connected buyers. For this reason we can simplify the notation and write $p_j(t)$ instead of $p_{ij}(t)$.

– *Edge dynamic*: When a buyer is connected to several sellers, it may decide to stop interacting with one of them, if the offered price is too high in comparison with the others. A buyer decides also to stop the transaction with one seller if the quantity of goods offered is too low. Formally

$$(i,j)\begin{cases}\text{cut} & \text{if } p_j(t) > \tau \min(p_l(t)) \text{ or } g_{ji}(t) < \epsilon, \quad l \in N_i^{out} \\ \text{not cut} & \text{, otherwise}\end{cases} \quad . \tag{4}$$

where τ and ϵ are a parameters.

Then, the state of each cell i at the next iteration $t+1$ is given by the balance of cash and goods,

$$c_i(t+1) = c_i(t) - \sum_{j \in N_i^{out}} c_{ij}(t) + \sum_{k \in N_i^{in}} c_{ki}(t) \quad . \tag{5}$$

$$g_i(t+1) = g_i(t) - \sum_{j \in N_i^{out}} g_{ij}(t) + \sum_{k \in N_i^{in}} g_{ki}(t) \quad . \tag{6}$$

3 GT-Model Simulator and Simulations Setup

The GT-model simulator is the tool that allows the user to interact with the settings and the evolution of the GT-model and to visualize and save the results. It is based on the general CAG formalism described in [8]. This tool, illustrated in Fig. 2, is composed of the following modules:

– **CAG Evolution:** This is the core of the architecture. It implements the dynamics of the GT-model.

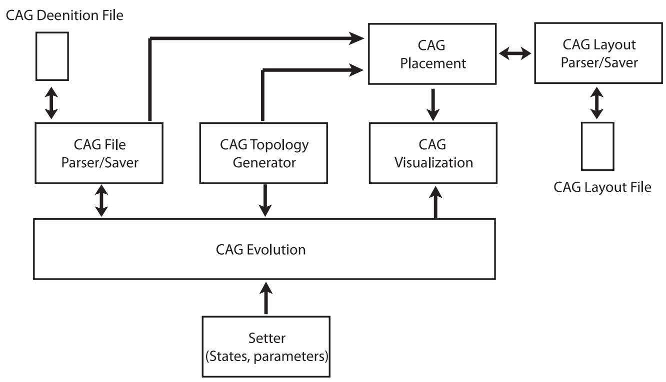

Fig. 2. The architecture of the GT-model simulator

- **Setter:** This module is in charge of initializing the state of each cell, the unit price of goods of each seller, the flow of cash and goods through the edges at any time iteration t.

  The initial conditions of the evolution are set by this module. We use two initial conditions which are EQ and RND initial conditions. The total amount of cash and goods in the whole GT-model are divided equally to the cells with EQ initial conditions and on the other hands, divided randomly with RND initial conditions. The initial unit prices of goods are either equals or randoms or chosen by the user.

  We choose the same value of λ and the same value of μ for all the cells.
- **CAG File and CAG Layout Parser/Saver:** The CAG Definition File contains all the information about the states of each cell, the flows transiting through each edge. The CAG Layout File contains all the (X, Y) coordinates of each cell on the visualization screen. These Parser/Saver modules parse and convert the Definition and Layout files to the data structure used by the CAG Evolution module. Vice versa they save the snapshot of the evolution at given time iteration to the Definition and Layout files.
- **CAG Topology Generator:** This module generates GT-model based on regular, random and scale-free-graph.
- **CAG Placement and CAG Visualization:** Before the visualization, the 2D coordinates of each cell are calculated by the CAG Placement module using a Force Directed Placement algorithm. Color scale are used by the CAG visualization module to visualize the state of each cell and the fluxes of cash and goods through each edge. Red color for the highest value, orange for the middle and green for the lowest value.

In addition to these modules, our implementation allows the user to modify the data of the model at run time. For instance a new link can be created, or the amount of cash or goods can be modified in a chosen cell.

We have performed experiments using the EQ and RND initial conditions on the following graph topologies: Erdös-Rényi random graph [9,13] (symmetric or not), scale-free graph [12,1,2,13] (symmetric or not), strongly connected graph (SCG) [10]. The values we choose for the parameters are $\lambda = 0.2$, $\mu = 0.3$, $\epsilon = 0.01$ and $\tau = 10$.

4 Results and Discussions

The evolution of the GT-model can be first described by a transient regime in which prices, goods and cash flows are time dependent and links can be cut. During this phase, it is observed that the cutting rule has the effect of make the Strongly Connected Components (SCC) of the graph emerge. The SCC [10] are the sub-graphs such that there is a return path between any pairs of nodes in the SCC. In other words, the GT-model is such that a link in G survives only if all the money or goods that flow through this edge has a path back to where it came from, even if this path is long. Figure 3 depicts the states of `erd_n30_m39_eq`

(a)

(b)

Fig. 3. Evolution of the `erd_n30_m39_eq` GT-model, during its transient regime. This topology is constructed by Erdös-Rényi algorithm with $n = 30$, $m = 39$ and simulated with EQ initial conditions (a) at $t = 0$ and (b) at $t = 100$.

GT-model at $t = 0$ and $t = 100$. We observe clearly at $t = 100$ the emergence of the following SCC: {1,18,28}, {14,19,23}, {4,6,29,27,21}.

Using the economical interpretation, each emerging SCC corresponds to an independent submarket in which the total amount of cash and goods is constant. When reaching the stationary state the unit price of goods inside each SCC is observed to converge to a uniform value. Let us denote this equilibrium price as p_e. The fluxes of cash and goods transiting through each cell are also in equilibrium. Each cell gives to its sellers the exact amount of cash received from its buyers

$$\lambda_i c_i = \sum_{j \in N_i^{in}} c_{ji} \tag{7}$$

Therefore, from (3) we have

$$p_e = \frac{\sum_{j \in N_i^{in}} c_{ji}}{\mu_i g_i} = \frac{\lambda_i c_i}{\mu_i g_i} \ . \tag{8}$$

Since p_e is observed to be independent of i in the steady state, we have $\mu_i g_i = p_e \lambda_i c_i$ for all i in the same sub-market. When all the cells make the same investment of cash and goods ($\lambda_i = \lambda$ and $\mu_i = \mu$), we can sum this relation over i and we obtain that the equilibrium price is the ratio between the total investment of cash and the total investment of goods

$$p_e = \frac{\lambda c_{tot}}{\mu g_{tot}} \ . \tag{9}$$

In other words the price is determined by a global supply (μg_{tot}) and demand (λc_{tot}) law.

We can write a mathematical equation for the distribution of cash and goods in a sub-market. We introduce a_{ij}, the element of the adjacency matrix A of the graph G and k_i^{out} the outdegree of the cell i). From the stationary assumption we have

$$\lambda_i c_i = \sum_{j \in N_i^{in}} c_{ji} = \sum_{j \in N_i^{in}} a_{ji} c_{ji} \tag{10}$$

and, since the price is uniform, we have from (1)

$$c_{ij} = \frac{\lambda_i c_i}{k_i^{out}} = \frac{\lambda_i c_i}{\sum_{\ell \in N_i^{out}} a_{i\ell}} \tag{11}$$

Therefore the values c_i obey

$$\lambda_i c_i = \sum_{j \in N_i^{in}} \frac{a_{ji}}{\sum_{\ell \in N_j^{out}} a_{j\ell}} \lambda_j c_j \tag{12}$$

Using in addition that $c_{tot} = \sum_i c_i$, we can solve analytically (12) for a symmetric graph ($a_{ij} = a_{ji}$). We find [13]

$$c_i = \frac{k_i^{in}}{m} c_{tot} \ . \tag{13}$$

Thus the amount of cash owned by each cell is proportional to the number of its connected buyers and inversely proportional to the market size defined by the number of edges m of the graph G. In a scale-free graph, the node degrees obey a power law distribution and the above relation shows that the wealth distribution follow a Zipf's law.

Compared with the results obtained during the simulations, the analytical results fit perfectly well the simulation results, as illustrated in Fig. 4 in case of symmetric graph G.

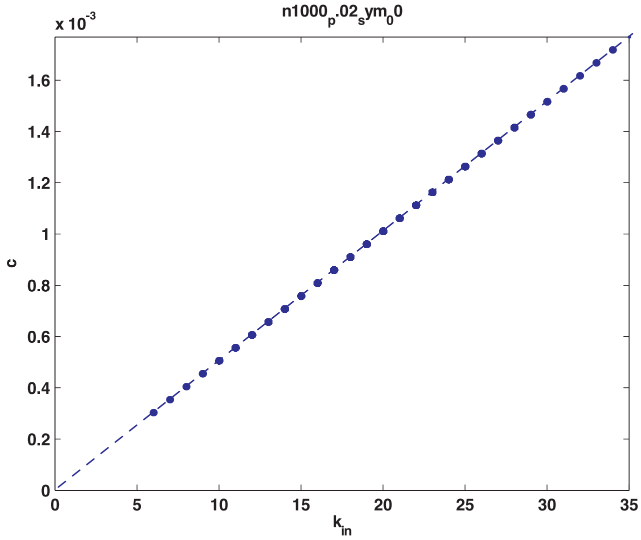

Fig. 4. The dots show the distribution of cash of a stationary GT-model based on symmetric random graph with $n = 1000$ and $m = 19790$ versus indegree k^{in}. The dotted line is the distribution of cash calculated analytically.

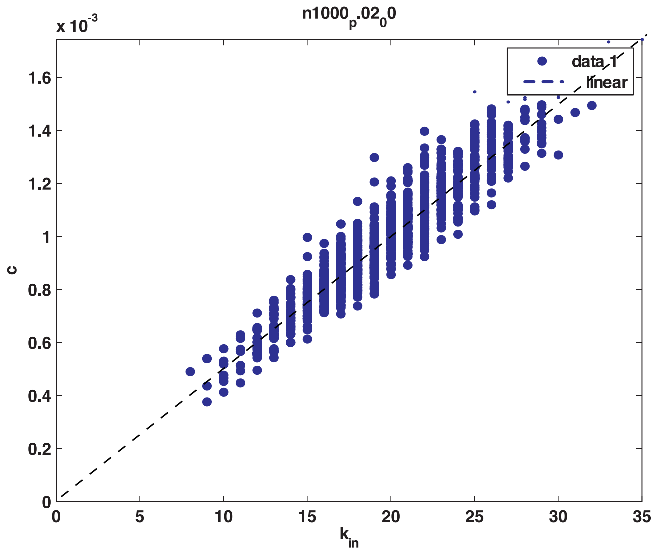

Fig. 5. The dots show the distribution of cash of a stationary GT-model based on asymmetric random graph with $n = 1000$ and $m = 19800$ (a) versus indegree k^{in}. The dotted line is the linear estimation of the distribution given by Eq. (13).

In the case of an asymmetric graph, one has to solve numerically (12). This is easily done for any given adjacency matrix, as (12) is a linear system of equation for c_i. Figure 5 shows the values of the fortune as a function of the indegree of the graph. We see that nodes with the same degree may have different amount of cash. However, we observe that these values are globally compatible with the prediction of (13). We believe that as the number of nodes increases the dispersion will reduce.

5 Conclusion and Future Work

In this work, we have studied the behavior of a particular cellular automaton on a graph (CAG) called Give and Take model (GT-model). This model simulates the exchange of cash against goods between the cells. At each time iteration t, each cell i gives cash to its outgoing neighbors j and in exchange takes goods from them. This relation is represented by an edge (i, j) of the graph G of the CAG. The graph topology also evolves: relations that become too expensive are progressively abandoned.

We found that during the transient regime, the strongly connected components of the graph emerge and form independent sub-markets.

In the stationary regime, and within each sub-market we observed that the unit price of goods becomes homogeneous and obeys a supply and demand law. By the simulations and analytic calculations, the amount of cash owned by each cell is proportional to the number of its connected buyers and inversely proportional to the size of the "market". Thus, the distribution of cash and goods in the whole automata depends only on the topology. For scale-free graph a power law for the wealth distribution is then observed.

From an application point of view, we plan to extend the model by allowing the production of goods and money (open systems) and by adding a work-salary market.

From the computer science point of view, we plan to analyze the performance of our model as an algorithm to detect the SCC. Further work are in progress to parallelize the CAG evolution algorithm and to standardize the simulator to allow the definition of more CAG models.

References

1. Albert, R., Barabási, A.-L.: Topology of evolving networks: local events and universality. Physical Review Letters, 5234 (2000)
2. Barabási, A.-L., Albert, R., Jeong, H.: Mean-Field Theory for Scale-Free Random Networks. Physica A, 173–187 (1999)
3. Bar-Yam, Y.: Dynamics of Complex Systems (2003)
4. Braha, D., Bar-Yam, Y.: The Statistical Mechanics of Complex Product Development: Empirical and Analytical Results. Microeconomics (2005)
5. Berge, C.: Graphs and hypergraphs (1976)
6. Caldarelli, G.: Scale-Free Networks. Oxford University Press, Oxford (2007)

7. Chopard, B., Droz, M.: Cellular Automata modeling of physical systems. Cambridge University Press, Cambridge (1998)
8. Chopard, B., Falcone, J.-L., Razakanirina, R.M., Hoekstra, A., Caiazzo, A.: On the Collision-Propagation and Gather-Update Formulations of a Cellular Automata Rule. In: Umeo, H., Morishita, S., Nishinari, K., Komatsuzaki, T., Bandini, S. (eds.) ACRI 2008. LNCS, vol. 5191, pp. 144–151. Springer, Heidelberg (2008)
9. Durrett, R.: Random Graph Dynamics (2006)
10. Hartmann, A.K., Weigt, M.: Phase Transitions in Combinatorial Optimization Problems (2005)
11. Mitchell, M., Newman, M.: Complex Systems Theory and Evolution. In: Encyclopedia of Evolution (2002)
12. Li, L., Alderson, D., Tanaka, R., Doyle, J.C., Willinger, W.: Towards a Theory of Scale-Free Graphs: Definition, Properties, and Implications (Extended Version). In: Internet Mathematics (2005)
13. Razakanirina, R.M.: Cellular automata over graph applied on financial and goods flows simulation. Master's thesis, University of Geneva, Computer Science department, Master thesis (2007)
14. Angeles Serrano, M., Boguna, M., Vespignani, A.: Patterns of dominant flows in the world trade web. In: Quantitative Finance Papers (2007)