Dynamic Processes on Graphs

Francesco Giuseppe Gillio
Politecnico di Torino

I. PROBLEM OVERVIEW

The research attempts to simulate dynamic processes on graphs to study the effect of initial conditions on dynamics. The simulator attempts to efficiently generate Erdős-Rényi random graph models, two-dimensional regular grid models, and three-dimensional regular grid models with `1000` nodes each.

Furthermore, the research attempts to establish a discrete event simulation flow according to a Poisson process, with variable lambda intensity depending on the degrees of freedom of the model nodes, to simulate a voter model on the network.

II. PROBLEM SOLVING

In the current section the report attempts to address, step by step, the different layers of the problem solving approach, such as the network models generation algorithms, the `FES` management and the simulator architecture.

A. *Erdős-Rényi random graph model generation algorithm*

The simulator implements the `G(n,p)` function to efficiently generate the Erdős-Rényi random graph model. The function structures the network according to a dictionary-type data structure, with name `N`. The network dictionary archives the node identification codes as keys and the node objects as values. For each node, the function establishes the voter's state as `+1` or `-1`, with variable probability depending on the input parameter. The function therefore implements a random process for generating edges between nodes according to a binomial model in which for each pair of nodes in the network an edge appears with probability `p = 10/n`. Finally, the simulator structures the `GC` (giant component) search function via the deep-first search (`DFS`) algorithm, set to recursively iterate over each node of the network to search for the subnetwork with the greater number of nodes. The function then returns the network as a dictionary-type data structure, with name `N`, containing only the nodes of the giant component.

B. *Two-dimensional regular grid model generation algorithm*

The simulator implements the `D2` function to efficiently generate the two-dimensional regular grid model. The function structures, as before, the network according to a dictionary-type data structure, with name `N`. The network dictionary archives the node identification codes as keys and the node objects as values. For each node the function establishes the voter's state as `+1` or `-1`, with variable probability depending on the input parameter. The function therefore implements a random process to generate the two-dimensional coordinates for each node of the network and a geometric process to generate edges between each pair of nodes as a function of the geometric distance.

C. *Three-dimensional regular grid model generation algorithm*

The simulator implements the `D3` function to efficiently generate the two-dimensional regular grid model. The function structures, as before, the network according to a dictionary-type data structure, with name `N`. The network dictionary archives the node identification codes as keys and the node objects as values. For each node the function establishes the voter's state as `+1` or `-1`, with variable probability depending on the input parameter. The function therefore implements a random process to generate the three-dimensional coordinates for each node of the network and a geometric process to generate edges between each pair of nodes as a function of the geometric distance.

D. *FES managemen*

The management of the `FES` occurs according to previous simulation architectures. The `FES` therefore archives a succession of wake-up events, each of which establishes the future of the following one according to a Poisson process with variable lambda depending on the degrees of freedom of the model nodes. Upon extraction of the wake-up event of the current node, the simulator operates the voter model.

E. *Simulator architecture*

The simulator, after generating the network according to the appropriate model, establishes the `FES` and initializes the first event by randomly selecting a network node at time 0. Subsequently, the simulator advances to discrete events by recursively calling the following functions:

- the `voter` model implementation function, set to simulate the Markov process of the voter model by uniformly and randomly selecting a neighbor of the node and copying the state accordingly. Furthermore, in case of uniformity of the states among the neighbors of the node, the function returns a random node of the network, otherwise the function returns a random node among the neighbors of the node.
- the node `wake-up` function, set to take the return node from the `voter` model implementation function, select the value of the lambda parameter according to the degrees of freedom of the node, generate a node wake-up time according to lambda and insert the wake-up Poisson event into the `FES`.

The simulator then ends the event loop upon reaching unanimity consensus among the states of the network nodes, i.e. after each node in the network appers with the same voter's state.

III. RESULTS

The research structures several simulations, each with different input parameters, to study the effect of initial conditions on the dynamics of the network. Please note that the research implements approximately `1000` simulations for each combination, to reach an accuracy score of `0.85` over a confidence level of `0.85`.

The simulator calculates the accuracy in relation to the most relevant metrics of the project, i.e. the probability of `+1` consensus and the average time to reach unanimity in consensus. Below, the research reports the various results obtained and tries to draw some interesting conclusions.

A. `G(1000,0.01)` *with state probability of* `0.51`

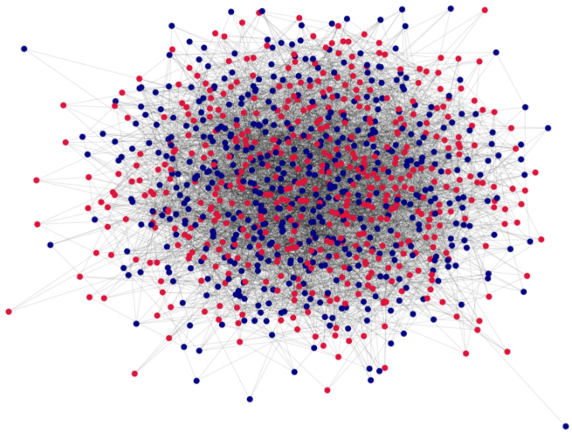

Figure 1

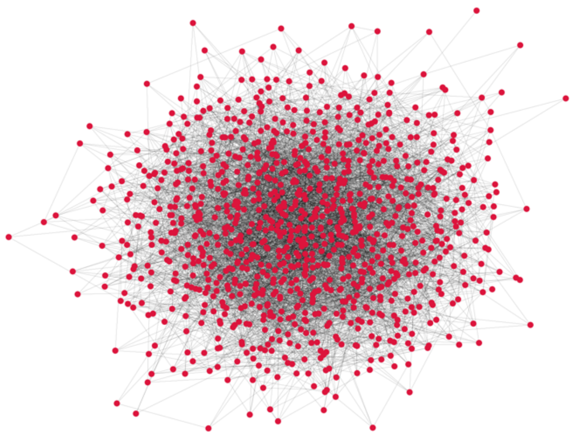

Figure 2

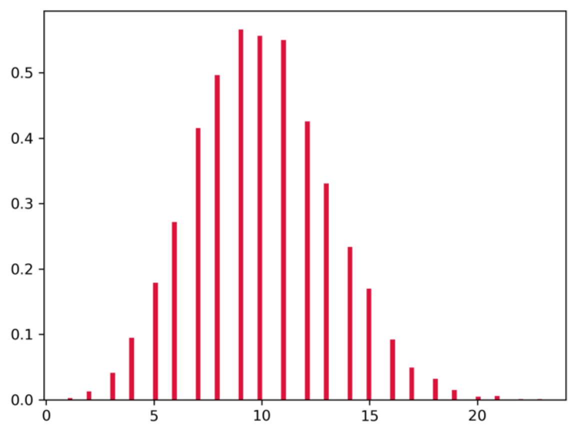

Figure 3

Figure 1 shows an initial configuration of the Erdős-Rényi random graph model `G(1000,0.01)`, with approximately `51%` of the nodes belonging to the `+1` state, and `49%` to the `-1` state. Figure 2 shows a final configuration of the Erdős-Rényi random graph model `G(1000,0.01)`, with unanimity consensus for the `+1` state. Figure 2 shows the most frequent final condition among the approximately `1000` simulations. Figure 3 shows the empirical distribution of the degrees of the Erdős-Rényi random graph model `G(1000,0.01)`.

B. `G(1000,0.01)` *with state probability of* `0.55`

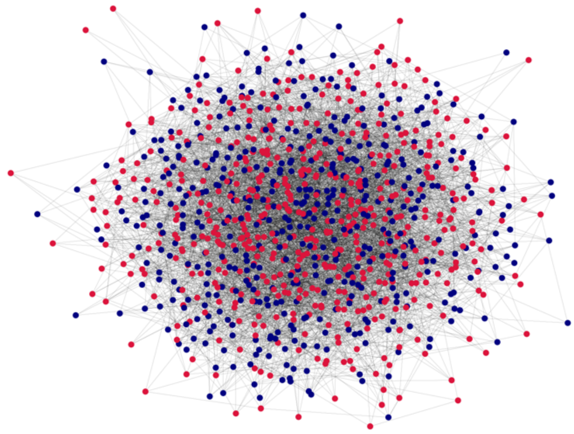

Figure 4

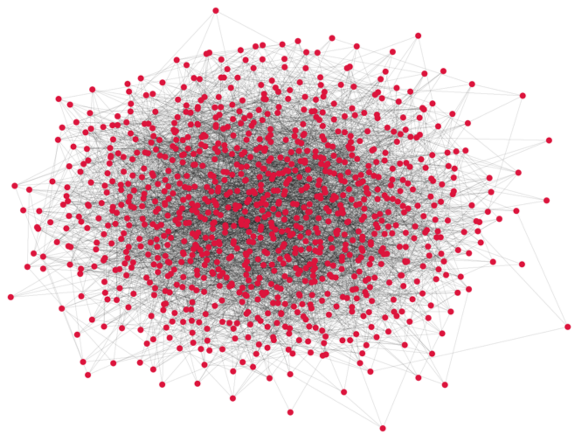

Figure 5

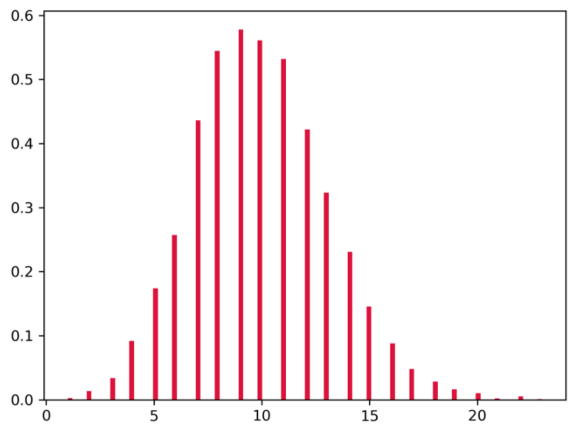

Figure 6

Figure 4 shows an initial configuration of the Erdős-Rényi random graph model `G(1000,0.01)`, with approximately `55%` of the nodes belonging to the `+1` state, and `45%` to the `-1` state. Figure 5 shows a final configuration of the Erdős-Rényi random graph model `G(1000,0.01)`, with unanimity consensus for the `+1` state. Figure 5 shows the most frequent final condition among the approximately `1000` simulations. Figure 6 shows the empirical distribution of the degrees of the Erdős-Rényi random graph model `G(1000,0.01)`.

C. $G(1000, 0.01)$ with state probability of 0.6

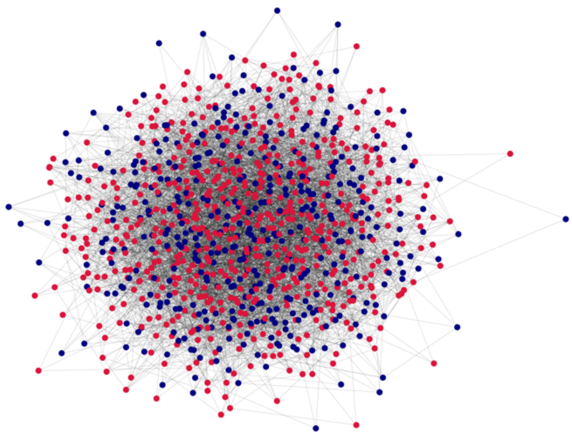

Figure 7

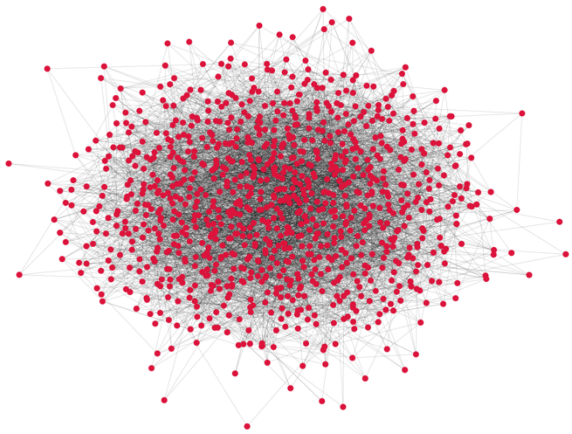

Figure 8

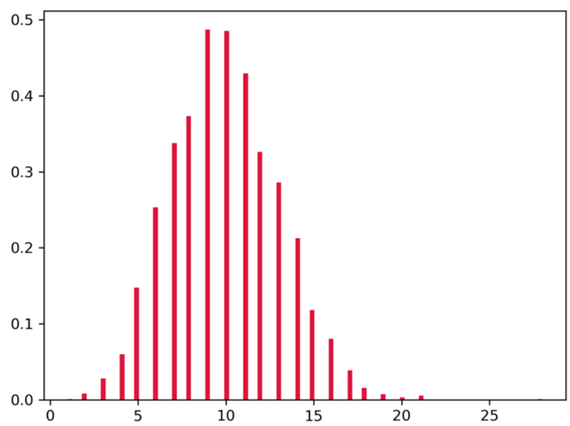

Figure 9

Figure 7 shows an initial configuration of the Erdős-Rényi random graph model $G(1000, 0.01)$, with approximately 60% of the nodes belonging to the $+1$ state, and 40% to the -1 state. Figure 8 shows a final configuration of the Erdős-Rényi random graph model $G(1000, 0.01)$, with unanimity consensus for the $+1$ state. Figure 8 shows the most frequent final condition among the approximately 1000 simulations. Figure 9 shows the empirical distribution of the degrees of the Erdős-Rényi random graph model $G(1000, 0.01)$.

D. $G(1000, 0.01)$ with state probability of 0.7

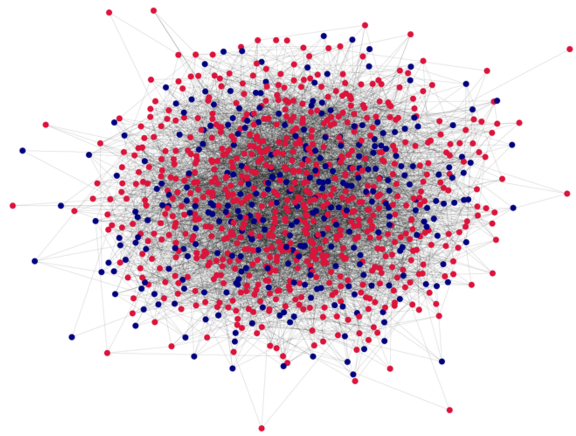

Figure 10

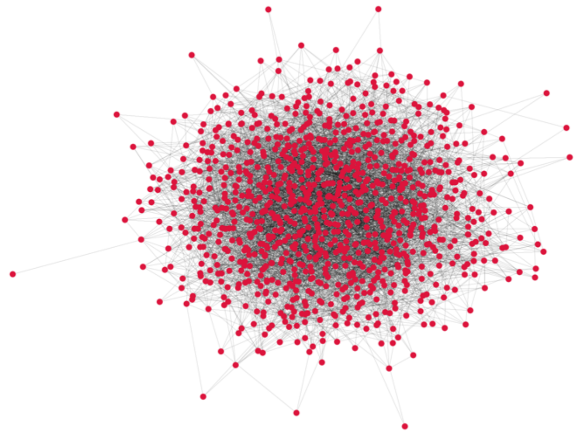

Figure 11

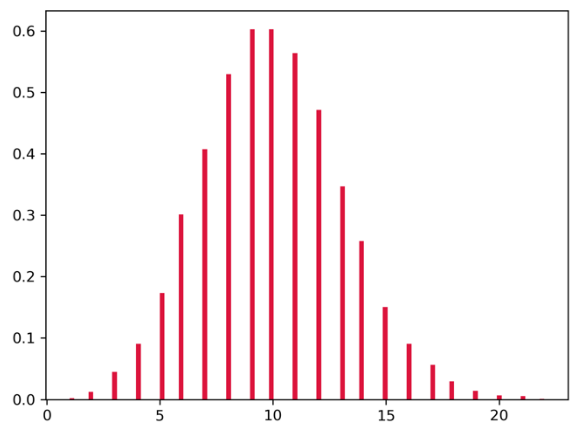

Figure 12

Figure 10 shows an initial configuration of the Erdős-Rényi random graph model $G(1000, 0.01)$, with approximately 70% of the nodes belonging to the $+1$ state, and 30% to the -1 state. Figure 11 shows a final configuration of the Erdős-Rényi random graph model $G(1000, 0.01)$, with unanimity consensus for the $+1$ state. Figure 11 shows the most frequent final condition among the approximately 1000 simulations. Figure 12 shows the empirical distribution of the degrees of the Erdős-Rényi random graph model $G(1000, 0.01)$.

E. 2D regular grid with state probability of 0.51

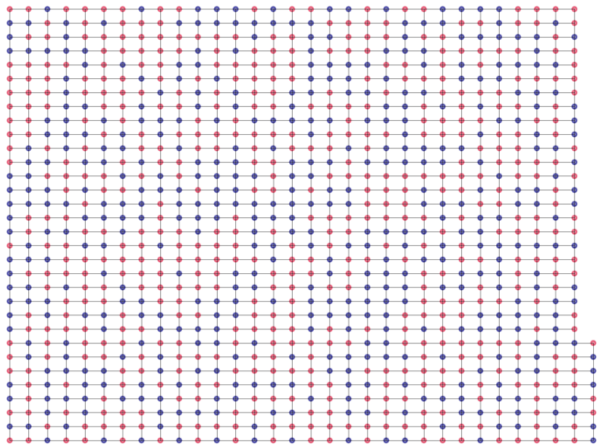

Figure 13

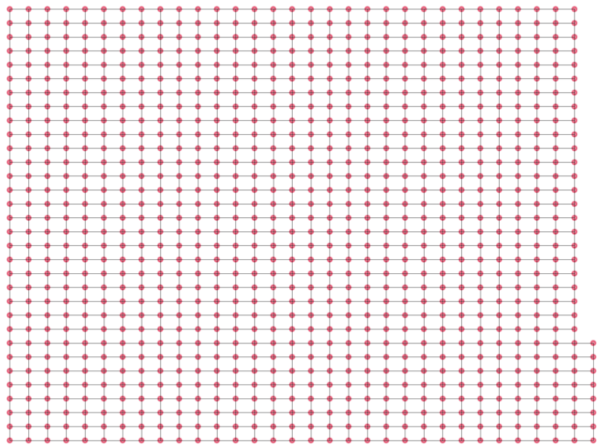

Figure 14

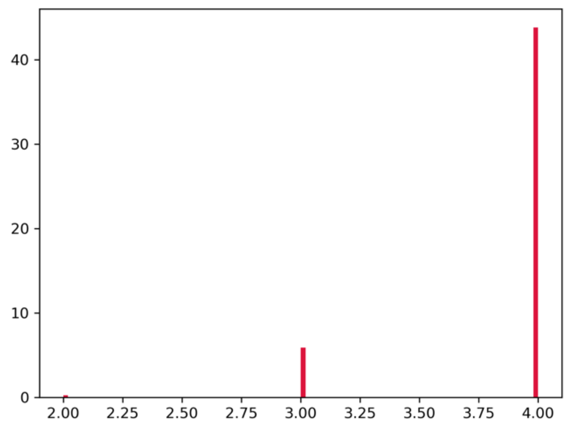

Figure 15

Figure 13 shows an initial configuration of the two-dimensional regular grid model, with approximately 51% of the nodes belonging to the +1 state, and 49% to the −1 state. Figure 14 shows a final configuration of the two-dimensional regular grid model, with unanimity consensus for the +1 state. Figure 14 shows the most frequent final condition among the 1000 simulations. Figure 15 shows the empirical distribution of the degrees of the two-dimensional regular grid model.

F. 3D regular grid with state probability of 0.51

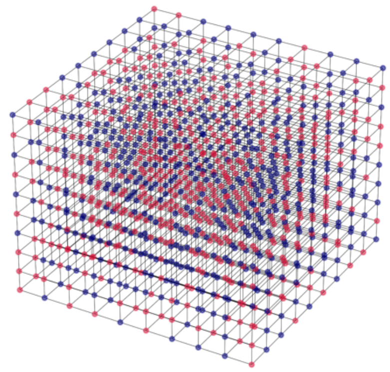

Figure 16

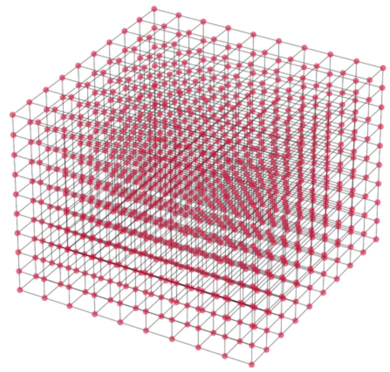

Figure 17

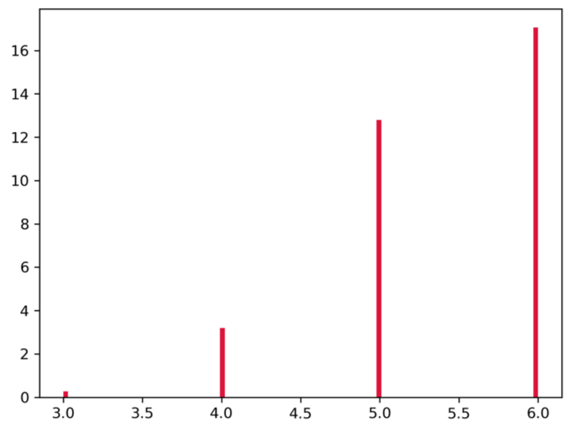

Figure 18

Figure 16 shows an initial configuration of the three-dimensional regular grid model, with approximately 51% of the nodes belonging to the +1 state, and 49% to the −1 state. Figure 17 shows a final configuration of the three-dimensional regular grid model, with unanimity consensus for the +1 state. Figure 17 shows the most frequent final condition among the 1000 simulations. Figure 18 shows the empirical distribution of the degrees of the three-dimensional regular grid model.

IV. DISCUSSION

The following table shows the summary results for the simulations on the *Erdős-Rényi random graph model*:

state probability	*average time*	*+1 consensus probability*
0.51	5153.50	0.55
0.55	4260.43	0.75
0.60	3545.15	0.90
0.70	1590.27	0.95

The following table shows the summary results for the simulations on *two-* and *three-dimensional regular grid models*:

model	*average time*	*+1 consensus probability*
2D	12457.55	0.50
3D	7562.16	0.55

After approximately `1000` simulations for each parameter combination, in terms of model type and initial distribution of node states, and almost 50 hours of processing, the research reaches a significant result: both the model, in particular the degrees of freedom of the model nodes, and the initial condition of the state distribution (state probability) significantly affect both the probability of reaching `+1` consensus and the average time to reach unitary consensus.

The results appear in line with theoretical predictions. In fact, the voter model represents a Markov process that changes the state of the nodes depending on the state of a random neighbor and, as a consequence of the random nature of the process, the initial condition statistically influences the result.

The degrees of freedom reflect the average number of connections between nodes in the network, and as connections increase, the spread of a dominant state occurs more quickly. Furthermore, in the presence of a dominant state, the probability that a random neighbor of a node belongs to this state results greater than the opposite case, thus favoring the diffusion of the same dominant state throughout the network. As the number of connections increases, and as the percentage of a dominant state in a network increases, the voter model statistically tends to consensus the dominant state faster and faster.

The results show in fact how as the initial percentage of nodes in the dominant state (the `+1` state) increases, and as the degrees of freedom of the model nodes increase, the probability of obtaining a unitary `+1` consensus consequently increases and the average time to reach unitary consensus decreases. The *Erdős-Rényi random graph model* shows a considerably greater number of degrees of freedom compared to the degrees of the *two-dimensional* and *three-dimensional regular grid model*, which in fact shows a significantly shorter average time to reach unitary `+1` consensus. Just as, again taking into account the number of connections between the nodes, the three-dimensional regular grid model shows an average time to reach unitary +1 consensus shorter than the two-dimensional regular grid model. Considering the *Erdős-Rényi random graph model*, as the initial percentage of nodes in the dominant state (the `+1` state) increases, the average time to reach unitary `+1` consensus decreases considerably as a consequence of the intrinsic aspect of the Makarov process of the voter model, in which the random choice of the neighbor obtains a greater probability of reaching a node with state `+1`.