

# Evolutionary Dynamics on Small-World Networks

Jan Rychtář and Brian Stadler

**Abstract**—We study how the outcome of evolutionary dynamics on graphs depends on a randomness on the graph structure. We gradually change the underlying graph from completely regular (e.g. a square lattice) to completely random. We find that the fixation probability increases as the randomness increases; nevertheless, the increase is not significant and thus the fixation probability could be estimated by the known formulas for underlying regular graphs.

**Keywords**—evolutionary dynamics, small-world networks.

## I. INTRODUCTION

Evolutionary dynamics has been traditionally studied in infinite homogenous, [1], infinite spatial, [2], populations. Recently, the dynamics was studied in finite, [3], and spatially structured populations (i.e. graphs), [4], [5, Chapter 8], [6], [7].

A graph can capture spatial, social and/or other structures of the population. Graphs can represent nearly everything we encounter in life, cities interconnected by highways, the national power grid, ecological structures, the social networks within which we communicate, etc. Many of these naturally occurring graphs can be modeled as 'small-world' networks, [8].

For the purpose of the dynamics, every vertex of a graph represents an individual. Individuals can place offspring into adjacent vertices. How often an individual is selected for reproduction is proportional to individual's fitness. The higher the fitness, the more likely it is that the individual will be selected.

For a given graph, it is important to know what happens when a mutant with a fitness  $r$  is introduced into a population of otherwise equal individuals of fitness 1. There are three possible scenarios:

- 1) A mutant population will eventually spread through the entire graph and will replace all of the original inhabitants,
- 2) The original population will recover from the mutant invasion and will eventually wipe out all mutants,
- 3) Both mutants and the original inhabitants stay in the population for an infinite amount time.

We are interested in the fixation probability  $\varrho$  of a mutant, i.e. a probability of the scenario 1). Note that since the dynamics is not deterministic, the probability is never 1, no matter how advantageous is the mutant. Also, it seems intuitive that the higher the fitness  $r$  of a mutant, the higher its fixation

J. Rychtář is with the Department of Mathematics and Statistics University of North Carolina at Greensboro Greensboro, NC 27402, USA, email: rychtar@uncg.edu.

B. Stadler is with the Department of Computer Science University of North Carolina at Greensboro Greensboro, NC 27402, USA, email: btstadle@uncg.edu.

The research was supported by the NSF grant 0634182. B. Stadler was supported by the UNCG Undergraduate Research Assistantship.

probability. However, fixation probability does not have to depend on  $r$  this way, [4].

The fixation probability for special graphs such as a complete graph, a square lattice, and a hexagonal lattice, is given by

$$\varrho = \frac{1 - 1/r}{1 - 1/r^N}, \quad (1)$$

where  $N$  is the size of the population, [4].

The dynamics on small-world networks, [9], and random graphs, [10], [11], were studied. A small-world network is a random graph where most vertices are not neighbors to each other, yet the majority of vertices can be reached from any other vertex in a relatively short path through the graph. Social networks, the Internet and national power grids all exhibit the characteristics of small-world networks, [9].

We produced a small-world network starting from a regular graph structure by randomly rewiring a gradually increasing percentage of edges. We then looked at how the fixation probability  $\varrho$  of mutants depends on the following three parameters:

- fitness  $r$  of a mutant,
- percentage of the changed vertices  $c$ ,
- and the underlying original regular graph structure.

## II. MATHEMATICAL BACKGROUND

Only undirected graphs,  $G = (V, E)$ , where  $V = \{0, 1, 2, \dots, N - 1\}$  is the set of vertices and  $E$  is the set of edges, were considered.

We regarded the dynamics as a Markov chain, [12]. If the mutants already inhabit vertices in the set  $C \subset V$ , then in the next step the mutants will inhabit either

- 1) a set  $C \cup \{j\}$ ,  $j \notin C$ , provided a) a vertex  $i \in C$  was chosen for a reproduction and b) it placed its offspring into  $j$ ; or
- 2) a set  $C \setminus \{i\}$ ,  $i \in C$ , provided a) a vertex  $j \notin C$  was selected for a reproduction and b) it placed its offspring into a vertex  $i$ ; or
- 3) a set  $C$ , provided an individual from  $C$  ( $V \setminus C$ ) replaces another individual from  $C$  ( $V \setminus C$ ).

The states  $\emptyset$  and  $V$  are the absorbing points of the dynamics. To determine the transitions probabilities of the above Markov chain, we have to determine a) the probability that a given vertex will be selected for a reproduction and b) the probability that, once selected, it places its offspring into another vertex.

Let an individual  $i$  have a fitness  $f_i \in \{1, r\}$ , where  $f_i = r$  means that  $i$  is a mutant. To be selected with a probability proportional to its fitness means that an individual  $i$  is selected for a reproduction with probability

$$s_i = \frac{f_i}{\sum_{j=0}^{N-1} f_j}. \quad (2)$$

We represented the graph by a matrix  $W = (w_{ij})$ , where  $w_{ij} = 0$ ,  $i$  and  $j$  are not connected, and  $w_{ij} = e_i^{-1}$ , if  $i$  and  $j$  are connected, where  $e_i$  is the number of edges coming out of a vertex  $i$ . In this notation,  $w_{ij}$  is the probability that (if selected for a reproduction) an individual  $i$  places an offspring into a vertex  $j$ .

Let  $P_C$  denote the probability of mutant fixation (i.e. mutants reaching the state  $V$ ) provided mutants currently inhabit a set  $C$ . The rules of the dynamics yield

$$P_C = \frac{r \sum_{i \in C} \sum_{j \notin C} w_{ij} P_{C \cup \{j\}}}{r \sum_{i \in C} \sum_{j \notin C} w_{ij} + \sum_{j \notin C} \sum_{i \in C} w_{ji} + \frac{\sum_{j \notin C} \sum_{i \in C} w_{ji} P_{C \setminus \{i\}}}{r \sum_{i \in C} \sum_{j \notin C} w_{ij} + \sum_{j \notin C} \sum_{i \in C} w_{ji}}} \quad (3)$$

with the “boundary conditions”

$$\begin{aligned} P_\emptyset &= 0, \\ P_G &= 1. \end{aligned}$$

The above system (3) of linear equations is unfortunately very large and very sparse. From any state  $C$ , one can go to at most  $N$  other states, i.e. each row of the transition matrix contains at most  $N$  non-zero elements. Yet, even the simplest graphs like a circle or a line contain of the order of  $N^2$  states that can be reached starting from a single mutant in any position  $i \in V$ . Moreover, once the graph is not a circle and not a tree, (3) consists of of the order of  $2^N$  equations, [13].

As a result, (3) can be seldom solved. In [4], authors abandoned (3) completely by restricting themselves to isothermal graphs, i.e. graphs satisfying

$$\sum_{j=0}^{N-1} w_{ji} = \sum_{j=0}^{N-1} w_{ji'}, i \neq i'.$$

For such graphs, (3) reduces due symmetries into a one dimensional random walk that corresponds to the Moran process, [14]. Examples of isothermal graphs are complete graphs, square lattice, hexagonal lattice, etc.

In [15], authors solved (3) for special values of  $r \approx 1$  and  $r \gg 1$ .

Our goal is to attack the dynamics for general  $r$  and general graphs. We used a Markov Chain Monte Carlo method, [16], in order to get the estimate of  $P_{\{i\}}$ ,  $i \in V$  and then calculate the fixation probability

$$\varrho = \frac{1}{N} \sum_{i=0}^{N-1} P_{\{i\}}.$$

### III. MCMC

#### A. Computing background

The MCMC simulation of the evolutionary dynamics had to:

- represent complicated graph structures,

- be able to quickly manipulate these graphs,
- generate random numbers.

C++ was chosen as the language to handle the simulation. It was primarily chosen due to available C++ libraries from <http://www.boost.org>, which can handle the requirements listed above. Below are the specific libraries used in development of the program:

- Boost date\_time libraries - to aid in seeding the random number generator,
- Boost graph libraries - for creation of undirected graphs and tracking the properties of each vertex,
- Boost random libraries - for creation of large amounts of near non-deterministic numbers. Specifically, the mt19937 generator was used which has a cycle of  $2^{19937} - 1$ .

#### B. Evolutionary dynamics

Below is a pseudo code representing the evolutionary dynamics on a given graph:

```
while (not (graph fully mutated or without mutants)) do
  select (a base vertex for reproduction)
  find (all neighboring vertices of base vertex)
  select (a neighboring vertex to be replaced)
  replace (neighboring vertex by offspring of base vertex)
end
```

The core of the process is the selection of the base vertex for reproduction. An individual labeled  $i$  is chosen for a reproduction with probability  $s_i$  given by (2). Once a base vertex is selected, all adjacent vertices are found. If a given base vertex has  $M$  neighbors, a specific neighbor is selected with a probability  $1/M$ .

#### C. Creation of small-world networks and random graphs

The procedure for creation of small-world networks follows [9]. We randomly rewire a given percentage of edges in the original graph. Loop back and parallel edges were not allowed. We also disregarded all disconnected graphs that could possibly result in this procedure (a disconnected graph has fixation probability 0 because mutants can never spread outside the component they were originally introduced to). Below is pseudo code representing the creation of these graphs:

```
while (less than  $c$  percent of original edges changed) do
  select(random base vertex)
  find (all neighboring vertices of base vertex)
  select (one neighboring vertex)
  destroy(edge between base and neighbor vertex)
  select(new random vertex)
  create(edge between base and new random vertex)
end
check(connected graph)
```

All selections here are not fitness related. If we had  $N$  vertices, one particular vertex was chosen with probability  $1/N$ . If we had  $M$  neighboring vertices to choose from, one

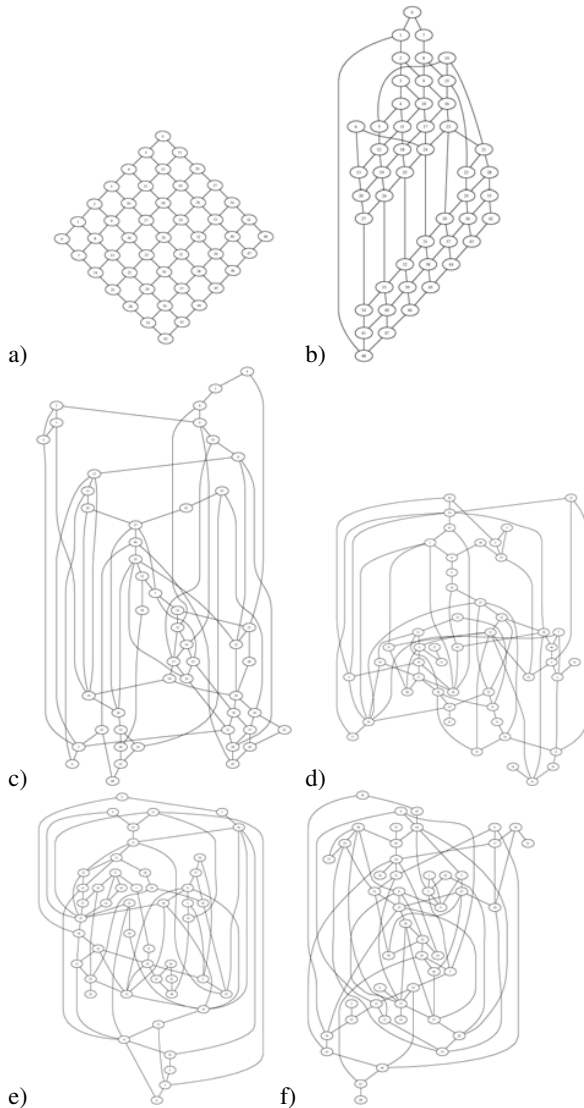


Fig. 1. Increase of randomness. a) Regular square lattice ( $c = 0$ ), b)  $c = 0.05$ , c)  $c = 0.25$ , d)  $c = 0.5$ , e)  $c = 0.75$ , f)  $c = 1$ .

particular vertex was selected with the probability  $1/M$ . The selection of the new random vertex had to be a bit more sophisticated. To assure that the resulting graph does not have loops or multiple edges, we had to exclude the base vertex itself as well as all vertices that were neighboring with the base vertex at the time of its selection. Figure 1 shows how a regular square lattice changes its structure as more and more edges are rewired.

#### D. Simulations

Evolutionary dynamics with mutant fitness  $r \in \{1 + j/10; j = 1, 2, \dots, 10\}$  were simulated. We did not simulate  $r > 2$  extensively since it is not biologically too relevant to have a mutant with a fitness significantly higher than the rest of the population.

For each fitness value a set of 20 graphs,  $\{G_i, i =$

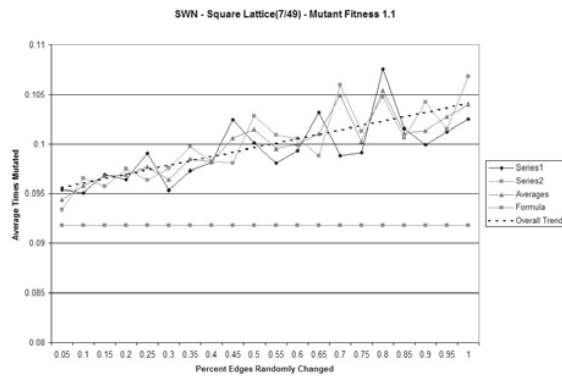


Fig. 2. Fixation probabilities for square lattice with 49 vertices and a relative mutant fitness 1.1.

$1, \dots, 20\}$  was produced using the above rewiring algorithm to randomly rearrange the edges of the base graph. Graph  $G_i$  had  $i \cdot 5\%$  of rewired edges. We then ran the MCMC simulation on each such graph and recorded the results. In the beginning, we performed  $10^6$  runs for each graph and every mutant fitness  $r$ . Since  $10^6$  runs took anywhere from 10 to 24 hours for one graph, the worst case time scenario was about 200 days using a single computer. No large scale computing environment was available.

After completing results for  $r = 1.1, r = 1.5$  and  $r = 2$  we looked at all of the results and it was determined that the results would not differ significantly if we stopped the simulations after  $10^5$  runs. Thus, the remaining results were generated using this smaller amount of runs.

#### IV. RESULTS

There are two fundamental results of the simulations:

- 1) the mutants do perform better in the changed graphs than in the original regular graphs;
- 2) the fixation probability of mutants in rewired graphs is never significantly above the value predicted by the formula (1).

Generally, the more edges were rewired the better the mutants performed. See Figure 2, where results are plotted for mutant fitness  $r = 1.1$  and compared to (1). Results for other mutant fitness in the range  $1.1 \leq r \leq 2$  were analogous.

As one can see from Figure 2, it is not always true that the more edges rewired the better the mutants perform. There are some minor fluctuations, cases when a slight increase of randomness actually decreased the fixation probability of mutants. The fluctuations flatten out as we consider different rewirings and taking averages over the rewirings.

We observed that the fixation probability of mutants in rewired graphs is never significantly above the value predicted by the formula (1). As we can see from Figure 3, one can effectively estimate the fixation probability in any random rewiring. The fixation probability was never more than .041 above the value predicted by the formula (1) (this was the case of  $r = 1.5$ ); respectively never more than 17% of the

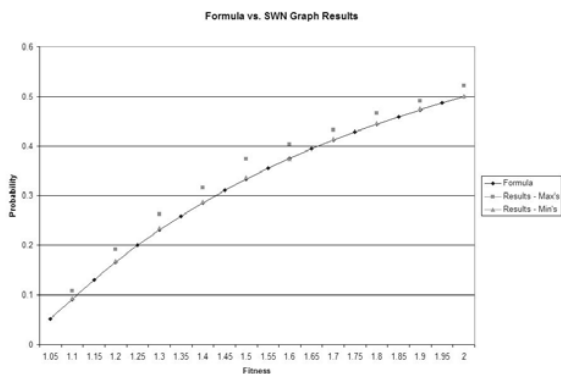


Fig. 3. Difference in fixation probabilities for square lattice with 49 vertices,  $r \in [1.1, 2]$  and  $c \in [0, 1]$ .

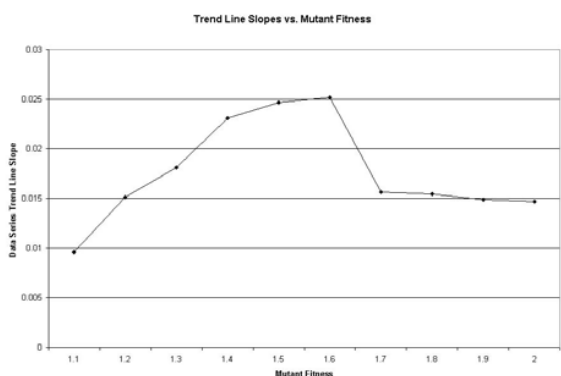


Fig. 4. Slopes of the trend lines as a function of mutant's fitness.

predicted value (the case of  $r = 1.1$ ), typically 9% above the predicted value.

We also noticed a behavior that we cannot explain. As already noted, the more edges we changed, the higher the fixation probability of mutants. As seen in Figure 2, the dependence is linear, i.e. for a given base graph, the fixation probability for a graph with  $c$  edges changed is given by

$$\varrho(c) = \varrho(0) + a + bc.$$

Figure 4 shows how the slope of the trend lines depends on the fitness  $r$ . The slopes are increasing for  $r$  between 1.1 and 1.6 (from 0.01 to 0.025) and then it sharply goes down to roughly 0.015 and stays around this value. Although it seems plausible that the effect of the graph structure will be weaker as mutants' fitness gets larger, we still do not know what is the exact cause of this sharp decline.

## V. DISCUSSION

We have studied the evolutionary dynamics on small world networks. We have shown that the mutants do perform better in the small-world network graphs than in regular graphs. The fixation probability is never below the level for a mutant with the same fitness in the base regular graph. However, it is never significantly above the level.

For smaller graphs, such as those representing computer networks, or power grids, it may be possible to predict how a computer virus will spread through a network or whether or not the failure of one power station will cause a cascading effect and in turn shut down the entire grid.

Further work needs to be done. If one wants to suppress a virus in a network what graph will best do this? The theoretical results from [4] as well as numerical simulations we performed suggest that the greater the variation of various parameters of the graph the better the mutants do. Among the characteristics to consider are vertex temperature (see [4] for the definition), path length, and clustering.

## ACKNOWLEDGMENT

The authors would like to thank the editors of the journal for valuable suggestions that helped to increase the quality of the paper.

## REFERENCES

- [1] Metz, J.A.J., Geritz, S.A.H., Mesza, G., Jacobs, F.J.A., van Heerwaarden, J.S., 1995. *Adaptive dynamics, a geometrical study of the consequences of nearly faithful reproduction*. Stochastic and spatial structures of dynamical systems (Amsterdam, 1995), 183–231, Konink. Nederl. Akad. Wetensch. Verh. Afd. Natuurk. Eerste Reeks, 45, North-Holland, Amsterdam, 1996.
- [2] Durrett, R., Levin, S., 1994. *The importance of being discrete (and spatial)*. *Teor. Pop. Biol.* **46** (3), 363–394.
- [3] Taylor, C., Fudenberg, D., Sasaki, A., et al., 2004. *Evolutionary game dynamics in finite populations*. *Bull. Math. Biol.* **66** (6), 1621–1644.
- [4] Lieberman, E., Hauert, C. Nowak, M.A., 2005. *Evolutionary Dynamics on Graphs*, *Nature* **433**, 312–316.
- [5] Nowak, M.A., 2006. *Evolutionary Dynamics: exploring the equations of life*. Harvard University Press.
- [6] Ebel, H., Bornholdt, S., 2002. *Coevolutionary games on networks*. *Phys. Rev. E* **66** (5): Art. No. 056118.
- [7] Ohtsuki, H., Lieberman, E., Hauert, C. Nowak, M.A., 2006. *A simple rule for the evolution of cooperation on graphs and social networks*, *Nature* **433**, 502–505.
- [8] Watts, D.J., 1999. *Small Worlds*, Princeton University Press.
- [9] Watts, D.J., Strogatz, S.H., 1998. *Collective dynamics of 'small world' networks*, *Nature* **393**, 440–442.
- [10] Erdős, P., Renyi, A., 1960. *On the evolution of random graphs*, *Publ. Math. Inst. Hungarian Acad. Sci.* **5**, 17–61.
- [11] Barabasi, A., Albert, R., 1999. *Emergence of scaling in random networks*, *Science* **286**, 509–512.
- [12] Norris, J.R. 1997. *Markov Chains*, Cambridge University Press.
- [13] Broom, M., 2007. Personal communication.
- [14] Moran, P.A.P., 1958. *Random processes in genetics*, *Proc. Camb. Phil. Soc.* **54**, 60–71.
- [15] Broom, M., Rychtář, J., Stadler, B., 2007. In preparation.
- [16] Smith, A.F.M., Roberts, G.O., 1993. *Bayesian computation via the Gibbs sampler and related Markov chain Monte Carlo methods*. *J. Roy. Statist. Soc. Ser. B* **55**, no. 1, 3–23.