

# FAST AND DETERMINISTIC COMPUTATION OF FIXATION PROBABILITY IN EVOLUTIONARY GRAPHS

Paulo Shakarian  
Network Science Center and  
Dept. of Electrical Engineering and Computer Science  
United States Military Academy  
West Point, NY  
email: paulo.shakarian@usma.edu

Patrick Roos  
Dept. of Computer Science  
University of Maryland  
College Park, MD  
email: roos@cs.umd.edu

## ABSTRACT

In evolutionary graph theory [1] biologists study the problem of determining the probability that a small number of mutants overtake a population that is structured on a weighted, possibly directed graph. Currently Monte Carlo simulations are used for estimating such *fixation probabilities* on directed graphs, since no good analytical methods exist. In this paper, we introduce a novel deterministic algorithm for computing fixation probabilities for strongly connected directed, weighted evolutionary graphs under the case of neutral drift, which we show to be a lower bound for the case where the mutant is more fit than the rest of the population (previously, this was only observed from simulation). We also show that, in neutral drift, fixation probability is additive under the weighted, directed case. We implement our algorithm and show experimentally that it consistently outperforms Monte Carlo simulations by several orders of magnitude, which can allow researchers to study fixation probability on much larger graphs.

## KEY WORDS

Modelling of Evolution, Network Science, Stochastic Models, Network Diffusion

## 1 Introduction

Evolutionary graph theory, introduced in [1] studies the problem of a mutant overtaking a population whose underlying structure is represented as a directed, weighted graph. This model has been applied to problems in evolutionary biology [2], physics [3], and game theory [4]. Most work with evolutionary graphs concerns computing the *fixation probability* - the probability that a certain subset of mutants overtakes the population. Although good analytical approximations are available for the undirected case [5, 6], these break down for directed, weighted graphs as shown in [7], even in the case of neutral drift<sup>1</sup>. As a result, most work dealing with evolutionary graphs rely on Monte Carlo simulations to approximate the fixation probability [8, 9, 10]. In this paper we develop a novel deterministic algorithm to compute fixation probability in the case of neutral drift in

<sup>1</sup>Neutral drift is the case where mutants and residents have the same fitness.

directed, weighted evolutionary graphs based on the convergence of “vertex probabilities” to the fixation probability as time approaches infinity. We then prove that the fixation probability computed in neutral drift is a lower bound for the fixation probability when the mutant is more fit than the resident, confirming the simulation observations of [11]. We also show that fixation probability under neutral drift is additive (even for weighted, directed graphs), which extends the work of [6] which proved this for undirected graphs. Further, we implemented our algorithm and show that it outperforms Monte Carlo simulations by several orders of magnitude. The paper is organized as follows. In Section 2 we review the original model of [1] and introduce the idea of “vertex probabilities.” In Section 3 we show how vertex probabilities can be used to find the fixation probability. This is followed by our experimental evaluation in Section 4. Finally, related work is discussed in Section 5.

## 2 Technical Preliminaries

The *Moran Process* [12] is a stochastic process used to model evolution in a well-mixed population. All the individuals in the population are either *mutants* or *residents*. The aim of such work was to determine if a set of mutants could take over a population of residents. In [1], evolutionary graph theory (EGT) is introduced, which generalizes the model of the Moran Process by specifying relationships between the  $N$  individuals of the population in the form of a directed, weighted graph. We re-introduce their model in this section. First, we define an evolutionary graph is defined as follows.

**Definition 1** (Evolutionary Graph (EG) [1]). *Given natural number  $N$ , set of individuals  $V = \{v_1, \dots, v_i, \dots, v_N\}$  and adjacency matrix  $W = [w_{ij}]$  s.t.  $\forall i, \sum_j w_{ij} = 1$  and  $w_{ii} = 0$ , the tuple  $\langle N, V, W \rangle$  is an **evolutionary graph (EG)**.*

Intuitively,  $w_{ij}$  is the weight of the edge from vertex  $v_i$  to  $v_j$ . In all literature on evolutionary graph theory known to the authors, the evolutionary graph is assumed to be *strongly connected* as defined below. We make the same assumption.

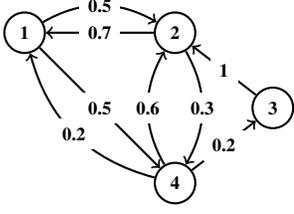


Figure 1: Example strongly connected, directed, and weighted graph.

**Definition 2** (Strongly Connected). An evolutionary graph  $\langle N, V, W \rangle$  is **strongly connected** if for all distinct vertices  $v_i, v_j \in V$ , there is a directed path from  $v_i$  to  $v_j$  such that for all edges in the path  $(v_a, v_b)$ ,  $w_{ab} > 0$ .

As all individuals in the population are either mutants or residents, we define a configuration of mutants and residents in the EG.

**Definition 3** (Configuration). Given evolutionary graph  $\langle N, V, W \rangle$ , a configuration  $C$  is a subset of  $V$  such that all individuals in  $C$  are mutants and all individuals in  $V - C$  are residents. The set  $2^V$  is the set of all configurations.

In [1], the authors specify a stochastic process known as the *Moran Process on Graphs* which we define below.

**Definition 4** (Graphical Moran Process (GMP) [1]). Given evolutionary graph  $\langle N, V, W \rangle$  and real number  $r > 0$  (known as the *relative fitness*), the **Graphical Moran Process (GMP)** is specified as follows. For any configuration  $C \subseteq V$ , some vertex  $v_i \in V$  is selected for birth. If  $v_i$  is in  $C$  it is selected with a probability  $\frac{r}{r \cdot |C| + |V - C|}$  and  $\frac{1}{r \cdot |C| + |V - C|}$  otherwise. Then vertex  $v_j$  is selected with probability  $w_{ij}$  for death and replaced by a clone of  $v_i$ . Hence, if  $v_i$  is a mutant, the new configuration is  $C \cup \{v_j\}$  and  $C - \{v_j\}$  otherwise.

Note if  $r = 1$ , we say we are in the special case of *neutral drift*. The following is an example of this process.

**Example 1.** Consider the evolutionary graph specified by Figure 1, which is directed, weighted, and strongly connected. Suppose vertex 1 in the graph is a mutant. Then, one possible sequence that leads to fixation is shown in Figure 2 and has the probability of  $2.93 \cdot 10^{-5}$  under neutral drift ( $r = 1$ ). For instance, to transition from the configuration labeled A to the configuration labeled B (where node 2 becomes a mutant as well) requires for node 1 to be picked for birth (with a probability of 0.25) and node 2 selected to die (which occurs with a probability of 0.5 by the edge weight). Hence, the transition to configuration B from A is 0.125.

Based on the definition of GMP, we will use the notation  $\Pr(C^{(t)})$  to refer to the probability of being in configuration  $C$  after  $t$  timesteps of the GMP. Perhaps the most widely studied problem in evolutionary graph theory is to determine the fixation probability. We define it formally below.

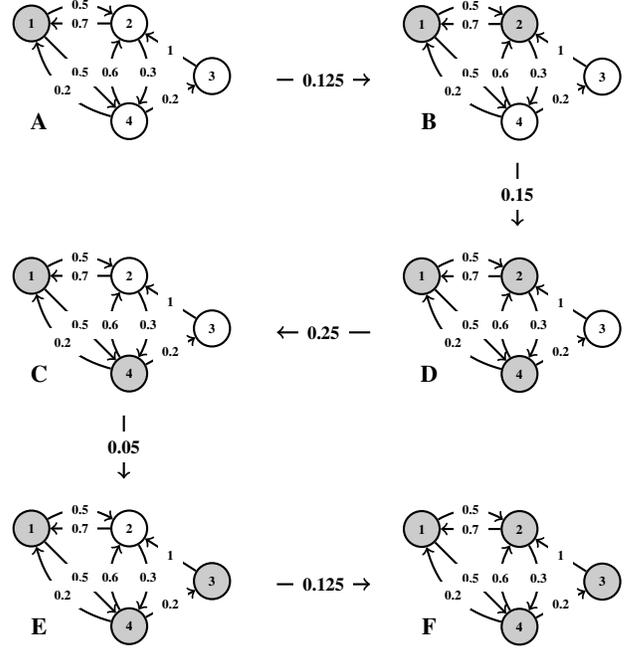


Figure 2: A sequence of mutant-resident configurations leading to fixation that happens with a probability of  $2.93 \cdot 10^{-5}$  under neutral drift. Mutant nodes are shaded. A is the initial state and F is the final absorbing state.

**Definition 5** (Fixation Probability). Given an evolutionary graph  $\langle N, V, W \rangle$ , real number  $r > 0$ , and configuration  $C \subseteq V$  the **fixation probability**,  $P_C$ , is  $\lim_{t \rightarrow \infty} \Pr(V^{(t)} | C^{(0)})$ .

Intuitively, this is the probability that an initial set  $C$  of mutants takes over the entire population. Similarly, we will use the term the *extinction probability*,  $\overline{P_C}$ , to be  $\lim_{t \rightarrow \infty} \Pr(\emptyset^{(t)} | C^{(0)})$ . If the graph is strongly connected, we have the following well-known result.

**Proposition 1.** Given EG  $\langle N, V, W \rangle$ , real number  $r > 0$ , and configuration  $C \subseteq V$ , if the EG is strongly connected, then under GMP we have  $P_C + \overline{P_C} = 1$

The above result essentially says that for a strongly connected graph, a mutant either fixates or becomes extinct. As [1] showed that an optimization problem related to computing fixation probability is NP-hard, much of the work on evolutionary graph theory relies on Monte Carlo simulations to calculate  $P_C$ . Algorithm 1 shows pseudo-code to find the  $P_C$  for the GMP using Monte Carlo simulations for the case of neutral drift. The time complexity is simply  $O(RT_{sim})$ , where  $R$  is the number of simulation runs executed (loop at line 2) and  $T_{sim}$  is the average time it takes for the evolutionary process to reach mutant extinction or fixation (the first nested loop at line 3).

---

**Algorithm 1** - Monte Carlo Simulation to Compute Fixation Probabilities

**Input:** Evolutionary Graph  $\langle N, V, W \rangle$ , configuration  $C \subseteq V$ , and natural number  $R > 0$ .

**Output:** Estimate of fixation probability of mutant.

---

```

1:  $x \leftarrow 0$  {x will count number of times the mutant fixates}
2: for  $i = 0 \rightarrow R$  do
3:   Set current configuration  $C^* = C$ 
4:   while While  $(C^* \neq V)$  or  $(C^* = \emptyset)$  do
5:     Select vertex  $v_i \in V$  with probability  $1/N$ 
6:     Select vertex  $v_j \in V$  with probability  $w_{ij}$ 
7:     If  $v_i \in C^*$  set  $C^* = C^* \cup \{v_j\}$ . Otherwise set
        $C^* = C^* - \{v_j\}$ 
8:   end while
9:   If  $C^* = V$  then  $x++$ 
10: end for
11: return  $x/R$  {the estimated fixation probability}

```

---

Further, based on the commonly-accepted definition of estimated standard error from statistics, we can obtain the estimated standard error for the solution returned by Algorithm 1.

**Proposition 2.** *The estimated standard error of the solution,  $S$  returned by Algorithm 1 is  $\sqrt{\frac{S(1-S)}{R-1}}$ .*

This work uses the idea of a *vertex probabilities* to create an alternative to Algorithm 1. A vertex probability is defined formally below.

**Definition 6** (Vertex Probability). *Given an evolutionary graph,  $\langle N, V, W \rangle$  and time  $t \geq 0$ , the vertex probability for  $v_i \in V$  is written  $\Pr(\mathbf{M}_i^{(t)})$  and defined as  $\sum_{C \in 2^V \text{ s.t. } v_i \in C} \Pr(C^{(t)})$ .*

Hence, the vertex probability is the probability that a given vertex at a certain time is a mutant. When we refer to the set vertex probabilities of each element of  $V$  at some time  $t$ , we shall use the term *vertex probability vector*. Viewing the probability that a specific vertex is a mutant at a given time has not, to our knowledge, been extensively studied before with respect to evolutionary graph theory (or the voter models in statistical physics). The key insight of this paper is that studying these probabilities sheds new light on the problem of calculating fixation probabilities. For example, it is easy to show the following relationship.

**Proposition 3.** *Let  $C$  be a subset of  $V$  and  $t$  be an arbitrary time point. Iff for all  $v_i \in C$ ,  $\Pr(\mathbf{M}_i^{(t)}) = 1$  and for all  $v_i \notin C$ ,  $\Pr(\mathbf{M}_i^{(t)}) = 0$ , then  $\Pr(C^{(t)}) = 1$  and for all  $C' \in 2^V$  s.t.  $C' \not\equiv C$ ,  $\Pr(C'^{(t)}) = 0$ .*

**Proof Sketch.** *Suppose, BWOC, there is some  $C' \not\equiv C$  where  $\Pr(C'^{(t)}) > 0$ . Clearly, there must exist some element  $v_i \in C$  that is not in  $C'$ . By Definition 6, this causes  $\Pr(\mathbf{M}_i^{(t)}) < 1$ , which is a contradiction. Going the*

*other direction, we need only to consider that  $\Pr(C^{(t)})$  is in the summand of each  $\Pr(\mathbf{M}_i^{(t)})$  associated only with a  $v_i \in C$ . ■*

We will also use random variables  $\mathbf{S}_i^{(t)}$  to denote the event that vertex  $v_i$  was selected for reproduction and  $\mathbf{R}_{ij}^{(t)}$  to denote the event of  $v_i$  replacing  $v_j$ . We will often use conditional probabilities. For example,  $\Pr(\mathbf{M}_i^{(t)} | C^{(0)})$  is the probability that  $v_i$  is a mutant given the initial set  $C$  of mutants. Throughout this paper, unless noted otherwise, all of our probabilities will be conditioned on  $C^{(0)}$ . We will drop it for ease of notation with the understanding that some set  $C$  of  $V$  were mutants at  $t = 0$ . Hence,  $\Pr(\mathbf{M}_i^{(t)}) = \Pr(\mathbf{M}_i^{(t)} | C^{(0)})$ . It is easy to verify that  $P_C > 0$  iff  $\forall v_i \in V$ ,  $\lim_{t \rightarrow \infty} \Pr(\mathbf{M}_i^{(t)}) > 0$ . Hence, in this paper, we shall generally assume that  $\lim_{t \rightarrow \infty} \Pr(\mathbf{M}_i^{(t)}) > 0$  holds for all vertices  $v_i$  and specifically state when it does not. As an aside, for a given EG, this assumption can be easily checked in polynomial time. Simply ensure for  $v_j \in V - C$  that exists some  $v_i \in C$  s.t. there is a directed path from  $v_i$  to  $v_j$ .

### 3 Directly Calculating Fixation Probability

Now that we have introduced the model and the idea of vertex probabilities we will show how to leverage this information in our algorithm to compute fixation probability. First, we show that as time approaches infinity, the vertex probabilities for all vertices converge to the fixation probability when the graph is strongly connected.

**Theorem 1.** *If the graph is strongly connected,  $\forall i$ ,  $\lim_{t \rightarrow \infty} \Pr(\mathbf{M}_i^{(t)}) = P_C$ .*

**Proof Sketch.** *As time approaches infinity, there are only two possible configurations of mutants - sets  $\emptyset$  and  $V$  corresponding with extinction and fixation respectively. By Definition 6, as time approaches infinity, the probability of any vertex being a mutant must be the equal to the fixation probability. Hence, the statement follows. ■*

Now let us consider how to calculate  $\Pr(\mathbf{M}_i^{(t)})$  for some  $v_i$  and  $t$ . For  $t = 0$ , where we know that we are in the state where only vertices in a given set are mutants, we need only appeal to Proposition 3 - which tells us that we assign a probability of 1 to all elements in that set and 0 otherwise. For subsequent timesteps, we have developed Theorem 2 shown next.

**Theorem 2.**  *$\Pr(\mathbf{M}_i^{(t)})$  equals*

$$\sum_{(v_j, v_i) \in E} \left( w_{ji} \cdot \Pr(\mathbf{M}_j^{(t-1)}) \cdot \Pr(\mathbf{S}_j^{(t)} | \mathbf{M}_j^{(t-1)}) - w_{ji} \cdot \Pr(\mathbf{M}_i^{(t-1)}) \cdot \Pr(\mathbf{S}_j^{(t)} | \mathbf{M}_i^{(t-1)}) \right) + \Pr(\mathbf{M}_i^{(t-1)})$$

*( $\mathbf{S}_i^{(t)}$  is true iff  $v_i$  is selected for reproduction at time  $t$ .)*

**Proof Sketch.** *Note we use the variable  $\mathbf{R}_{ji}^{(t)}$  is true iff  $v_j$  replaces  $v_i$  at time  $t$ . First we show that  $\Pr(\mathbf{M}_i^{(t)}) =$*

$\Pr(\mathbf{M}_i^{(t-1)} \wedge \bigwedge_{(v_j, v_i) \in E} \neg \mathbf{S}_j^{(t)} + \sum_{(v_j, v_i) \in E} \Pr(\mathbf{S}_j^{(t)} \wedge \mathbf{R}_{ji}^{(t)} \wedge \mathbf{M}_j^{(t-1)}) + \sum_{(v_j, v_i) \in E} \Pr(\mathbf{S}_j^{(t)} \wedge \neg \mathbf{R}_{ji}^{(t)} \wedge \mathbf{M}_i^{(t-1)})$  by the original model. Then, by the definition of conditional probability, we get  $\Pr(\mathbf{M}_i^{(t-1)} \wedge \bigwedge_{(v_j, v_i) \in E} \neg \mathbf{S}_j^{(t)}) = \Pr(\mathbf{M}_i^{(t-1)}) \cdot (1 - \sum_{(v_j, v_i) \in E} \Pr(\mathbf{S}_j^{(t)} | \mathbf{M}_i^{(t-1)}))$ . Then, by further manipulating the probability axioms as well as Bayes Theorem, we obtain  $\Pr(\mathbf{S}_j^{(t)} \wedge \mathbf{R}_{ji}^{(t)} \wedge \mathbf{M}_j^{(t-1)}) = w_{ji} \cdot \Pr(\mathbf{M}_j^{(t-1)}) \cdot \Pr(\mathbf{S}_j^{(t)} | \mathbf{M}_j^{(t-1)})$  as well as  $\Pr(\mathbf{S}_j^{(t)} \wedge \neg \mathbf{R}_{ji}^{(t)} \wedge \mathbf{M}_i^{(t-1)}) = (1 - w_{ji}) \cdot \Pr(\mathbf{M}_i^{(t-1)}) \cdot \Pr(\mathbf{S}_j^{(t)} | \mathbf{M}_i^{(t-1)})$ . After showing all of these items, we obtain the statement of the theorem through algebraic manipulation. ■

Although finding  $\Pr(\mathbf{S}_i^{(t)} | \mathbf{M}_j^{(t-1)})$  may be computationally intractable in practice, the good news is that for neutral drift ( $r = 1$ ), these conditional probabilities are trivial - specifically, we have  $\Pr(\mathbf{S}_i^{(t)}) = 1/N$  for all  $i$  and this is independent of the current set of mutants or residents in the EG. Hence, we can simplify  $\Pr(\mathbf{M}_i^{(t)})$  as follows.

**Corollary 1.** Under  $r = 1$ ,  $\Pr(\mathbf{M}_i^{(t)})$  equals

$$\frac{1}{N} \sum_{(v_j, v_i) \in E} w_{ji} \cdot (\Pr(\mathbf{M}_j^{(t-1)}) - \Pr(\mathbf{M}_i^{(t-1)})) + \Pr(\mathbf{M}_i^{(t-1)})$$

Studying evolutionary graph theory under neutral drift was a central theme in work such as [6, 7, 11] as it provides an intuition on the effects of network topology on mutant spread. We shall focus on neutral drift in this paper as well. This special case also allows us to strengthen the statement of Theorem 1 to a necessary and sufficient condition - showing that when the probabilities of all nodes are equal, then we can determine the fixation probability.

**Theorem 3.** When  $r = 1$ , if for some time  $t$ ,  $\forall i$ , the value  $\Pr(\mathbf{M}_i^{(t)})$  is the same, then  $\Pr(\mathbf{M}_i^{(t)}) = P_C$ .

*Proof Sketch.* Consider  $\Pr(\mathbf{M}_i^{(t)}) = \Pr(\mathbf{M}_i^{(t-1)}) + \frac{1}{N} \sum_{(v_j, v_i) \in E} w_{ji} \cdot (\Pr(\mathbf{M}_j^{(t-1)}) - \Pr(\mathbf{M}_i^{(t-1)}))$  when for  $t - 1$ ,  $\forall i, j$  we have  $\Pr(\mathbf{M}_j^{(t-1)}) = \Pr(\mathbf{M}_i^{(t-1)})$ . Clearly, in this case, the value for  $\Pr(\mathbf{M}_i^{(t)}) = \Pr(\mathbf{M}_i^{(t-1)})$ . As the probabilities of all vertices was the same at  $t - 1$ , they remain so at  $t$ . Therefore, in this case,  $\lim_{t \rightarrow \infty} \Pr(\mathbf{M}_i^{(t)}) = \Pr(\mathbf{M}_i^{(t)})$ . By Theorem 1, this equals  $P_C$ . ■

Therefore, under neutral drift, we can determine fixation probability when the equation of Corollary 1 causes all  $\Pr(\mathbf{M}_i^{(t)})$ 's to be equal. We can also use Corollary 1 to find bounds on the fixation probability for some time  $t$  by the following result.

**Theorem 4.** For any time  $t$ , under neutral drift ( $r = 1$ ),  $P_C \in [\min_i \Pr(\mathbf{M}_i^{(t)}), \max_i \Pr(\mathbf{M}_i^{(t)})]$ .

*Proof Sketch.* For any time  $t$ , under neutral drift ( $r = 1$ ),  $P_C \leq \max_i \Pr(\mathbf{M}_i^{(t)})$ .

We show that for each time step  $t$ ,  $\max_i \Pr(\mathbf{M}_i^{(t-1)}) \geq \max_i \Pr(\mathbf{M}_i^{(t)})$ . Hence, by showing that, for any time  $t'$ , we have  $\max_i \Pr(\mathbf{M}_i^{(t')}) \geq \lim_{t \rightarrow \infty} \max_i \Pr(\mathbf{M}_i^{(t)})$  which

by allows us to apply Theorem 1 and obtain the statement of this theorem. Let  $v_i$  be the vertex that at time  $t$  becomes achieves the greatest increase in vertex probability. At time  $t$ . The rest follows by Corollary 1, and the fact that  $\max_j \Pr(\mathbf{M}_j^{(t-1)}) \geq \Pr(\mathbf{M}_j^{(t-1)})$  Which gives us the upper bound. The second part of the proof, which is used for the lower bound, mirrors this part. ■

Under neutral drift, we can show that fixation probability is additive for disjoint sets. A similar result was proved for a special case (that is, undirected evolutionary graphs) in [6]. However, our proof differs from theirs in that we leverage Corollary 1.

**Theorem 5.** When  $r = 1$  for disjoint sets  $C, D \subseteq V$ ,  $P_C + P_D = P_{C \cup D}$ .

*Proof Sketch.* Consider some time  $t$  and vertex  $v_i$ . Clearly, by Corollary 1,  $\Pr(\mathbf{M}_i^{(t)})$  can be expressed as a linear combination of the form  $\sum_{v_j \in V} (C_j \cdot \Pr(\mathbf{M}_j^{(0)}))$  where  $C_j$  is a coefficient. We note that these coefficients are the same regardless of the initial configuration of mutants that  $\mathbf{M}_i^{(t)}$  is conditioned on. Hence,  $\Pr(\mathbf{M}_i^{(t)} | C^{(0)})$  is this positive function with  $\Pr(\mathbf{M}_j^{(0)}) = 1$  if  $v_j \in C$  and 0 otherwise (see Proposition 3). Hence, for disjoint  $C, D$ , for any  $v_i \in V$ , we have  $\Pr(\mathbf{M}_i^{(t)} | C^{(0)}) + \Pr(\mathbf{M}_i^{(t)} | D^{(0)}) = \Pr(\mathbf{M}_i^{(t)} | (C \cup D)^{(0)})$ . The statement follows. ■

In [11], the author observes experimentally (through simulation) that the fixation probability computed with neutral drift appears to be a lower bound on the fixation probability calculated with a mutant fitness  $r > 1$ . Here, we prove this to be true mathematically.

**Theorem 6.** For a given set  $C$ , let  $P^{(1)}(C)$  be the fixation probability under neutral drift and  $P^{(r)}(C)$  be the fixation probability calculated using a mutant fitness  $r > 1$ . Then,  $P^{(1)}(C) \leq P^{(r)}(C)$ .

*Proof Sketch.* After introducing some notation,<sup>2</sup> we show first, by the rules of dynamics, we show that if a some time period  $t$ , the probability distribution over mutant configurations is  $\mathbf{I}$ , mutant fitness  $r$ , and the transition functions used to reach the next time step are  $\chi_+^{(r)}, \chi_-^{(r)}$ , and all

<sup>2</sup>**Proof Setup.** We define an *interpretation*,  $\mathbf{I} : 2^V \rightarrow [0, 1]$  as probability distribution over mutant configurations. Hence, for some  $\mathbf{I}$  we have  $\sum_{C \in 2^V} \mathbf{I}(C) = 1$ . Next, we define a transition function that maps configurations of mutants to probabilities,  $\chi : 2^V \rightarrow [0, 1]$  where for any  $C \in 2^V$ ,  $\sum_{C' \in 2^V} \chi(C, C') = 1$ . We will use  $\chi_+$  and  $\chi_-$  to indicate if the transition is made with a mutant being selected for birth ( $\chi_+$ ) or resident ( $\chi_-$ ). Hence, for some  $C \in V$  and  $v \notin C$ ,  $\chi_-(C, C \cup \{v\}) = 0$  and  $\chi_+(C \cup \{v\}, C) = 0$ . Hence, for all  $C \in 2^V$ ,  $\sum_{C' \in 2^V} (\chi_+(C, C') + \chi_-(C, C')) = 1$ . If the transition function is based on birth-death and computed with some  $r > 1$ , then we will write it as  $\chi_+^{(r)}, \chi_-^{(r)}$  respectively. If computed with  $r = 1$ , then we write  $\chi_+^{(nd)}, \chi_-^{(nd)}$  respectively. For some  $C \in 2^C$ , let  $\text{inc}(C)$  be the set of all elements  $D \in 2^V$  s.t.  $|D| \geq |C|$  and  $\chi_+(C, D) > 0$ . For some  $C \in 2^C$ , let  $\text{dec}(C)$  be the set of all elements  $D \in 2^V$  s.t.  $|D| \leq |C|$  and  $\chi_-(C, D) > 0$ . Given set  $C \subseteq V$ , we will use  $P_C^{(r)}$  to denote the probability of fixation given initial set of mutants  $C$  where the value  $r$  is used to calculate all transition probabilities.

Table 1: Example of Algorithm 2 for Example 2.

$T$	0	1	2	...	49	50
$v_1$	1.0	0.775	0.629	...	0.1865	0.1864
$v_2$	0.0	0.125	0.175	...	0.1859	0.1859
$v_3$	0.0	0.000	0.006	...	0.1848	0.1849
$v_4$	0.0	0.125	0.206	...	0.1870	0.1868
$\tau$	0.5	0.388	0.311	...	0.0011	0.0009

subsequent transitions are computed using the same dynamics with neutral drift, then the fixation probability is:

$\mathcal{P}(\mathbf{I}, r) = \sum_{C \in 2^V} \mathbf{I}(C) \cdot \left( \sum_{D \in \text{inc}(C)} (\chi_+^{(r)}(C, D) \cdot P_D^{(1)}) + \sum_{D \in \text{dec}(C)} (\chi_-^{(r)}(C, D) \cdot P_D^{(1)}) \right)$ . Then, by algebraic manipulation, we show that for some  $r \leq r'$ , for all  $C, D \in 2^V$ , we have  $\chi_+^{(r)}(C, D) \leq \chi_+^{(r')}(C, D)$  and  $\chi_-^{(r)}(C, D) \geq \chi_-^{(r')}(C, D)$ . From Theorem 5, we know that given some  $C \in 2^V$ , for all pairs  $D, D'$  where  $D \in \text{inc}(C)$  and  $D' \in \text{dec}(C)$ , we have  $P_D^{(1)} \geq P_{D'}^{(1)}$ . We then prove that given interpretation  $\mathbf{I}$ , for some  $r > 1$ ,  $\mathcal{P}(\mathbf{I}, r) \geq \mathcal{P}(\mathbf{I}, 0)$ , from which the theorem follows. ■

### 3.1 The Algorithm

Now we have shown all the necessary properties of vertex probabilities to create our algorithm. Algorithm 2 shows pseudo-code to compute the fixation probability using our method. As described above, our method has found the exact fixation probability when all the probabilities in  $\mathbf{Pr}(\mathbf{M}^{(t)})$  (represented in the pseudo-code as the vector  $\mathbf{p}$ ) are equal. We use Theorem 4 to provide a convergence criteria based on value  $\epsilon$ , which we can prove to be the tolerance for the fixation probability. Next, we show the time complexity and approximation guarantee of the algorithm, which follows from the results of this section. We also provide an example of how it works in Example 2.

**Proposition 4.** • Where  $T_{\text{sol}}$  is the number of time steps to convergence (based on  $\epsilon$ ) and  $K$  is the average in-degree of the vertices in the evolutionary graph,  $O(T_{\text{sol}}NK)$  is the time complexity of Algorithm 2.

- Algorithm 2 returns the fixation probability  $P_C$  within  $\pm\epsilon$ .

**Example 2.** Consider the scenario introduced in Example 1. Suppose we decide to use Algorithm 2 to compute the fixation probability with  $\epsilon = 0.001$ . Table 1 shows the vertex probability vector at each time step, along with the value for  $\tau$  from Algorithm 2. Hence, at 50 timesteps, the algorithm returns a fixation probability of 0.186.

**Algorithm 2** - Our Novel Solution Method to Compute Fixation Probabilities

**Input:** Evolutionary Graph  $\langle N, V, W \rangle$ , configuration  $C \subseteq V$ , natural number  $R > 0$ , and real number  $\epsilon \geq 0$ .  
**Output:** Estimate of fixation probability of mutant.

- 1:  $p_i$  is the  $i$ th position in vector  $p$  corresponding with vertex  $v_i \in V$ .
- 2: Set  $p_i = 1$  if  $v_i \in C$  and  $p_i = 0$  otherwise.
- 3:  $\mathbf{q} \leftarrow \mathbf{p}$  { $\mathbf{q}$  will be  $\mathbf{p}$  from the previous time step.}
- 4:  $\tau \leftarrow 1$
- 5: **while**  $\tau > \epsilon$  **do**
- 6:   **for**  $v_i \in V$  **do**
- 7:      $sum \leftarrow 0$
- 8:      $\mathbf{m} \leftarrow \{v_j \in V | w_{ji} > 0\}$
- 9:     **for**  $v_j \in \mathbf{m}$  **do**
- 10:        $sum = sum + w_{ji} \cdot (\mathbf{q}_j - \mathbf{q}_i)$
- 11:     **end for**
- 12:      $\mathbf{p}_i \leftarrow \mathbf{q}_i + 1/N \cdot sum$
- 13:   **end for**
- 14:    $\mathbf{q} \leftarrow \mathbf{p}$
- 15:    $\tau \leftarrow (1/2) \cdot (\max p - \min p)$
- 16: **end while**
- 17: **return**  $(\min p) + \tau$

In the above result, Since  $T$  depends on a desired tolerance, we cannot compare our algorithm's performance to monte-carlo simulation without the right termination condition. Thus in our later experiments we first find fixation probabilities using monte-calro simulation and then find the number of timesteps  $T$  that it takes our solution method to find a fixation probability within standard error of the simulation method.

It is easy to show that the expected number of mutants as time approaches infinity is equal to  $P_C \cdot N$  - as  $P_C$  is the probability of being in the state where all the vertices are mutants and  $N$  is the population (as the only other possible state as time approaches infinity is extinction - which has no mutants). For any time step  $t$ , the expected number of mutants is  $\sum_{v_i \in V} \mathbf{Pr}(\mathbf{M}_i^{(t)})$  which follows directly from Definition 6. Hence, returning the average vertex probability for a sufficiently large value of  $t$  may also provide a good approximation. We also note, that as the vertex probabilities converge, the standard deviation of the  $p$  vector in Algorithm 2 could be a potentially faster convergence criteria. Note that using standard deviation of  $p$  and returning the average vertex probability would no longer provide us of the guarantee in Proposition 4, however it may provide good results in practice. The modifications to the algorithm would be as follows: line 15 would be  $\tau \leftarrow \text{st.dev}(p)$  and line 17 would be **return**  $\text{avg}(p)$ . We will refer to this as *Algorithm 2 with alternate convergence criteria* or *Algorithm 2-ACC* for short.

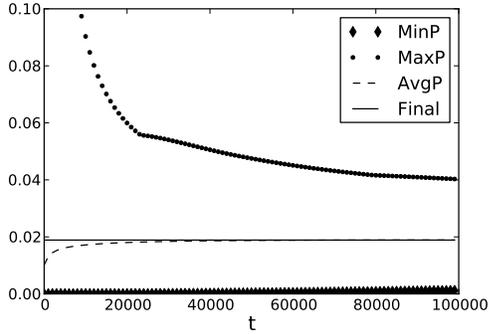


Figure 3: Convergence of the minimum (MinP), maximum (MaxP), and average (AvgP) of vertex probabilities towards the final fixation probability as a function of our algorithm’s iterations  $t$  for a graph of 100 nodes.

## 4 Experimental Evaluation

Our novel method for computing fixation probabilities on strongly connected directed graphs allows us to compute near-exact fixation probabilities within a desired tolerance. The time complexity of our method is highly dependent on how fast the vertex probabilities converge. In this section we experimentally evaluate how the vertex probabilities in our algorithms converge. We also provide results from comparison experiments to support the claim that *Algorithm 2-ACC* finds adequate fixation probabilities order of magnitudes faster than Monte Carlo simulations (*Algorithm 1*). All algorithms were implemented in Python and run on a 2.33GHz Intel Xeon CPU.

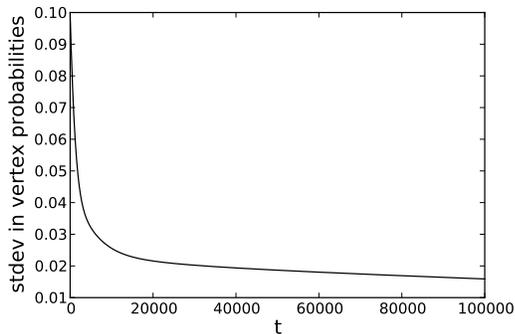


Figure 4: Standard deviation of vertex probabilities as a function of our algorithm’s iterations for the same 100 node graph of Figure 3.

### 4.1 Convergence of Vertex Probabilities

We ran our algorithms to compute fixation probabilities on randomly weighted and strongly connected directed graphs in order to experimentally evaluate the convergence of the vertex probabilities. We generated the graphs to be scale free using the standard preferential attachment growth

model [13] and randomly assigned an initial mutant node. We replaced all edges in the graph given by the growth model with two directed edges and then randomly assigned weights to all the edges.

Figure 3 shows the convergence of the minimum, maximum, and the average of vertex probabilities towards the final fixation probability value for a small graph of 100 nodes. We can observe that the average converges to the final value at a logarithmic rate and much faster than the minimum and maximum vertex probability values. This suggests that while *Algorithm 2-ACC* does not give the same theoretical guarantees as *Algorithm 2*, it is much preferable for speed since the minimum and maximum vertex probabilities take much longer to converge to the final solution than the average. The fact that the average of the vertex probabilities is much preferable as a fast estimation of fixation probabilities is supported by the logarithmic decrease of the standard deviation of vertex probabilities (see Figure 4). Convergences for other and larger graphs are not shown here but are qualitatively similar to the relative convergences shown in the provided graphs.

### 4.2 Speed Comparison to Monte Carlo Simulation

In order to compare our method’s speed compared to the standard Monte Carlo simulation method, we must determine how many iterations our algorithm must be run to find a fixation probability estimate comparable to that of *Algorithm 1*. Thankfully, as we have seen, we can get a standard error on the fixation probability returned by *Algorithm 1* as per Proposition 2. While we did not theoretically prove anything about how smoothly fixation probabilities from our methods approach the final solution, the convergences of the average and standard deviation as shown above strongly suggest that estimates from our method approach the final solution quite gracefully. In fact, in the following experiments once our method has arrived at a fixation probability estimate within the standard error of simulations, the estimate never again fell outside the window of standard error (although the estimate did not always approach the final estimate monotonically). This is in stark contrast to Monte Carlo simulations, from which estimations can vary greatly before the method has completed enough single runs to achieve a good probability estimate.

We generated a number of randomly weighted and strongly connected directed graphs of various sizes on which we compare our solution method to Monte Carlo approximation of fixation probabilities. The graphs were generated as in our convergence experiments. For each graph of a different size, we generate a number of different initial mutant configurations. We found fixation probabilities both using Monte Carlo estimation with 2000 simulation runs and our direct solution method, terminating when we have reached within the standard error of the Monte Carlo estimation. Since the average vertex probability proved to be such a good fast estimate of the true fixation probability, we used *Algorithm 2-ACC*.

Table 2: Experiment Results. For each graph size, shows average number of timesteps needed for each single simulation in the Monte Carlo estimation to reach extinction or fixation (Avg  $T_{sim}$ ), and the average number of iterations our solution algorithm must be run to get a fixation probability within the standard error of simulations (AVG  $T_{sol}$ ).

N	Avg $T_{sim}$	$T_{sol}$
100	19577.49	119.33
300	190425.88	584.38
500	857506.55	3240.2
700	946080.54	152.86
900	2216117.2	11052.83

Tables 2 and 3 show results from these experiments, including the average measured values for  $T_{sim}$  (average timesteps to extinction or fixation for simulations),  $T_{sol}$  (average timesteps for our solution method to get within standard error of Monte Carlo estimation), and actual seconds taken for each method implemented. Figure 5 shows the speedup our solution provides over Monte Carlo simulation. Here speedup is defined as the ratio of the time it takes for simulations to complete over the time it takes our algorithm to find a fixation probability within the standard deviation.

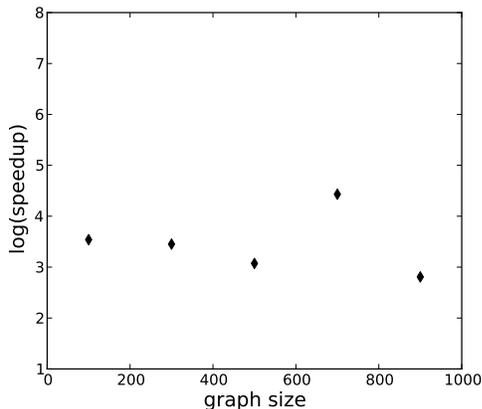


Figure 5: Average speedup (on a log scale) for finding fixation probabilities achieved by our algorithm vs Monte Carlo simulation for graphs of different sizes.

We can observe from our experiments that computing fixation probabilities using Monte Carlo simulations showed to be a very time-expensive process, highlighting the need for faster solution methods as the one we have presented. Especially for larger graph sizes, the time complexity of our solution to achieve similar results to Monte Carlo simulation has shown to be orders of magnitude smaller than the standard method.

Table 3: Experiment Results. For each graph size, shows the average seconds taken for the Monte Carlo method (Avg Secs Sim) and our solution method (Avg Secs Sol) to get a fixation probability within the standard error of simulations. Also shows the average speedup our algorithm provides.

N	Avg Secs Sim	Avg Secs Sol	Avg Speedup
100	3876.69	1.12	3461.68
300	41325.49	14.60	2829.74
500	206088.44	174.29	1182.48
700	220996.6	8.18	27025.63
900	507439.37	790.69	1385347.44

## 5 Related Work

While most work dealing with evolutionary graphs rely on Monte Carlo simulation, there are some good analytical approximations for the undirected case based on the degree of the vertices in question. In [5], the authors use the mean-field approach to create these approximations for the undirected case. In [6], the authors derive an exact analytical result for the undirected case in neutral drift, which agrees with the results of [5]. They also show that fixation probability is additive in that case (a result which we extend in this paper using a different proof technique). However, the results of [7] demonstrate that mean-field approximations break down in the case of weighted, directed graphs. That work is followed by [11] which also studies weighted, directed graphs, but does so by using Monte Carlo simulation. In [8] the authors derive exact computation of fixation probability through means of linear programming. However, that approach requires an exponential number of both constraints and variables and is intractable. In [10], the authors present a technique for speeding up Monte Carlo simulations by early termination. However, our algorithm differs in that it does not rely on simulation at all and provides a deterministic result. Our method totally avoids simulation and instead leverages properties of the model. Recently, in [14], the authors study the related problem of determining the probability of fixation given a single, randomly placed mutant in the graph where the vertices are “islands” and there are many individuals residing on each island in a well-mixed population. They use quasi-fixed points of ODE’s to obtain an approximation of the fixation probability and performed experiments with a maximum of 5 islands (vertices) containing 50 individuals each. It is unclear if their methods can be applied to the problem presented in this paper. In [15, 16] the authors present a generalized frameworks for social network diffusion. These works primarily focus on diffusion models that are monotonic - where the number of vertices with a certain property increases at each time step. While [15] does show how to adjust their framework for non-monotonic models, they only do so for a finite number of time steps (fixation probability is in the limit of time).

## 6 Conclusion

In this paper, we presented a novel deterministic algorithm for quickly computing fixation probability in directed, weighted strongly connected evolutionary graphs under neutral drift, which we prove to be a lower bound for the case where a mutant is more fit than the rest of the population. In our experiments, we showed our approach to outperform Monte Carlo simulations, which are currently used in most evolutionary graph theory research, by several orders of magnitude. We also show that under neutral drift, fixation probability is additive - showing optimal substructure in this case.

Our algorithm relied on the convergence of the “vertex probabilities” - the probability of an individual being a mutant at a certain time. As our experiments demonstrated that this convergence generally occurs rapidly, it is a tempting conjecture that the time to convergence is polynomial in the size of the graph. If this is the case, we suspect that fitness-based selection of the reproducing individual in the network may be a source of complexity in this problem. A more complete complexity analysis of evolutionary graph problems may still be in order.

An important limitation of our algorithm is that it is limited to strongly connected graphs. However, we believe that our algorithm can be used in solutions to general graphs by breaking the graph into its strongly connected components and considering transition edges between these in the computation of the overall fixation probabilities. Such an extension is an immediate goal for future work.

## Acknowledgements

P.S. is supported by ARO grant 611102B74F.

P.R. is supported by ONR grant W911NF0810144.

The opinions in this paper are those of the authors and do not necessarily reflect the opinions of the funders.

## References

- [1] E. Lieberman, C. Hauert, M. A. Nowak, Evolutionary dynamics on graphs, *Nature* 433 (7023) ,2005, 312–316.
- [2] P. A. Zhang, P. Y. Nie, D. Q. Hu, F. Y. Zou, The analysis of bi-level evolutionary graphs., *Biosystems* 90 (3) 2007, 897–902.
- [3] V. Sood, T. Antal, S. Redner, Voter models on heterogeneous networks, *Physical Review E* 77 (4) 2008, 041121.
- [4] J. M. Pacheco, A. Traulsen, M. A. Nowak, Active linking in evolutionary games, *Journal of Theoretical Biology* (3) 2006, 437–443.
- [5] T. Antal, S. Redner, V. Sood, Evolutionary dynamics on degree-heterogeneous graphs, *Physical Review Letters* 96 (18) 2006, 188104.
- [6] M. Broom, C. Hadjichrysanthou, J. Rychtar, B. T. Stadler, Two results on evolutionary processes on general non-directed graphs, *Proceedings of the Royal Society A* 466 (2121) 2010, 2795–2798.
- [7] N. Masuda, H. Ohtsuki, Evolutionary dynamics and fixation probabilities in directed networks, *New Journal of Physics* 11 (3) 2009, 033012.
- [8] J. Rychtář, B. Stadler, Evolutionary dynamics on small-world networks, *International Journal of Computational and Mathematical Sciences* 2 (1).
- [9] M. Broom, J. Rychtář, An analysis of the fixation probability of a mutant on special classes of non-directed graphs, *Proceedings of the Royal Society A* 464 2009, 2609–2627.
- [10] V. C. Barbosa, R. Donangelo, S. R. Souza, Early appraisal of the fixation probability in directed networks, *Physical Review E* 82 (4) 2010, 046114.
- [11] N. Masuda, Directionality of contact networks suppresses selection pressure in evolutionary dynamics, *Journal of Theoretical Biology* 258 (2) 2009, 323 – 334.
- [12] P. Moran, Random processes in genetics, *Mathematical Proceedings of the Cambridge Philosophical Society* 54 (01) 1958, 60–71.
- [13] H. Jeong, Z. Neda, A. L. Barabasi, Measuring preferential attachment in evolving networks, *EPL (Europhysics Letters)* 61 (4) 2003, 567.
- [14] B. Houchmandzadeh, M. Vallade, The fixation probability of a beneficial mutation in a geographically structured population, *New Journal of Physics* 13 (7) 2011, 073020.
- [15] D. Kempe, J. Kleinberg, E. Tardos, Maximizing the spread of influence through a social network, in: *KDD '03: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM, New York, NY, USA, 2003, pp. 137–146.
- [16] P. Shakarian, V. Subrahmanian, M. L. Sapino, Using generalized annotated programs to solve social network optimization problems, in: M. Hermenegildo, T. Schaub (Eds.), *Technical Communications of the 26th International Conference on Logic Programming*, Leibniz International Proceedings in Informatics (LIPIcs), Dagstuhl, Germany, 2010, pp. 182–191.