

# Approximating Spatial Evolutionary Games using Bayesian Networks

Anonymous Author(s)  
Submission Id: 178

## ABSTRACT

Evolutionary Game Theory is an application of game theory to evolving populations of organisms. Of recent interest are EGT models situated on structured populations or spatial evolutionary games. Due to the complexity added by introducing a population structure, model analysis is usually performed through agent-based Monte-Carlo simulations. However, it can be difficult to obtain desired quantities of interest from these simulations due to stochastic effects. We first define a framework for modeling spatial evolutionary games using Dynamic Bayesian Networks that capture the underlying stochastic process. The resulting Dynamic Bayesian Networks can be queried for quantities of interest by performing exact inference on the network. Taking inspiration from moment-closure approximation techniques, we then propose a method for producing approximations of the spatial evolutionary game through the truncation of the corresponding DBN. This method generalizes mean-field and pair approximations in the literature for spatial evolutionary games and we show that a special case of the method can be used to derive the differential equations for pair approximation. Furthermore, we show empirical results demonstrating the capability of the method to obtain much better accuracy than pair approximation with respect to stochastic simulations.

## KEYWORDS

Spatial Evolutionary Games, Bayesian Networks, Moment Closure

### ACM Reference Format:

Anonymous Author(s). 2021. Approximating Spatial Evolutionary Games using Bayesian Networks. In *Proc. of the 20th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2021), London, UK, May 3–7, 2021*, IFAAMAS, 9 pages.

## 1 INTRODUCTION

Evolutionary Game Theory (EGT) was initially developed to model biological evolution [18] but has found additional use in research on the evolution of cultural phenomena [8], and a variety of multi-agent systems topics [2, 22, 28, 29, 35]. EGT models provide a framework for modeling the time evolution of a population of agents that interact with each other through strategic games whose outcome determines each individual's evolutionary fitness. These models typically disregard any game-theoretic assumptions of rationality on individual agents and instead let individuals reproduce or change strategies stochastically based on a population update rule. A central tenet of these models is the idea that individuals that obtain a higher fitness are more likely to reproduce than those that obtained a lower fitness. When modeling cultural evolution, this implies that behaviors that give individuals a higher fitness are

more likely to spread through a population over time. While far from an exact description of human interactions, EGT models can be used to find trends that capture essential characteristics of the modeled interactions [7, 9].

In applications such as modeling cultural evolution, we might be interested in quantities such the proportions of each strategy or type of individual in the population and how fast the proportions change over time. These quantities have traditionally been studied using systems of differential equations commonly termed as evolutionary game dynamics [5] that approximate the interactions between agents in a population [33]. These models make the assumption that the population is well-mixed so that each agent is likely to interact with any other agent in the population. However, real world populations are rarely well-mixed and have relationships that would be better described as a social network. This spatial component adds a new level of complexity that is not easily addressed using existing game dynamic approaches. Attempts have been made to extend game dynamic models to spatial populations through techniques such as pair approximation [12, 16]. While these extensions may not be very accurate with respect to the underlying stochastic model [13, 15, 25, 31], there is still a great deal of interest in using them to obtain qualitative insights into model behavior [11, 12, 19, 24]. Alternatively, one can rely on agent-based stochastic simulations [1] to obtain insights into quantities of interest. However, there are cases such as EGT models with multiple equilibria where it is difficult to obtain the desired quantities of interest from the model using agent-based simulations due to stochastic effects. Beyond these specific situations, agent based simulations come with their own limitations in that they are difficult to validate [17] and may need to be repeated many times to reduce variability in results obtained [21].

In this paper, we propose a framework for the exact modeling of spatial evolutionary games using a Dynamic Bayesian Network (DBN) [6], thus making the whole toolbox of probabilistic inference algorithms applicable to such stochastic games [10]. Subsequently, we develop a method for producing approximations of stochastic spatial evolutionary games through the truncation of the corresponding DBN. The approximation method we propose generalizes existing mean-field techniques and their extensions in past literature [32] and we show that the framework can be used to obtain, as special cases, Bayesian Networks that can be used to derive the differential equations for existing pair approximation methods [12]. The power of our methodology is that it allows for a flexible framework for the exploration of approximation techniques such as moment closure (see Section 2) and higher order approximations beyond pair approximation that allow for better accuracy with respect to the underlying stochastic model. Finally, we provide preliminary empirical results illustrating the potential of our approach in modeling stochastic simulations and its advantages over existing approximations in the literature.

## 2 RELATED WORK

Past research on the approximation of stochastic evolutionary dynamics has mainly focused on the investigation of different systems of evolutionary dynamics. These dynamics are a continuous time approximation of the original discrete time process specified by the evolutionary game. The resulting systems of differential equations can be viewed as population-level models as they model the evolutionary game by analyzing the time evolution of the proportion of agents playing each strategy in the population  $p_i$ . For evolutionary games defined on a well-mixed population, it is simple to derive these dynamics by using the master equation that corresponds to the Markov process specifying the underlying microscopic dynamics [33]. Since the population is well mixed, the fitness of an individual is calculated from the proportion of each strategy in the population. As they rely on population level averages, these type of approximations are usually called mean-field approximations.

For spatial evolutionary games, the above techniques do not produce a closed system of equations. When applying the master equation to a spatial evolutionary game to find the time evolution of the proportion of a strategy in the population, we arrive at equations that depend on higher order quantities. Equations that specify the time evolution of the proportion of agents playing each strategy  $p_i$  in the population depend on the proportion of pairs of agents  $p_{ij}$  playing different strategy pairs in the population. In turn, equations that specify the time of evolution of pairs  $p_{ij}$  will depend on the proportion of triples  $p_{ijk}$ . This leads to a hierarchy of equations defined up to proportions of groups of agents the size of the entire population:

$$\begin{aligned} \dot{p}_i &= F(p_i, p_{ij}) \\ \dot{p}_{ij} &= G(p_i, p_{ij}, p_{ijk}) \\ \dot{p}_{ijl} &= H(p_i, p_{ij}, p_{ijk}, p_{ijkl}) \\ &\vdots \end{aligned} \quad (1)$$

These systems of equations are intractable to solve given a large enough population size. Consequently, there is much work in past literature [12, 16, 23, 26, 27, 31, 32] in which higher order proportions are approximated using lower order proportions. This idea of approximating higher order terms by lower order terms is an approximation technique known as moment closure [20]. Pair approximation uses a specific case of moment closure in which triplet terms are approximated using single and pair terms. For example, [12] defines a pair approximation where second order conditional probabilities are approximated using first order conditional probabilities as follows:

$$\begin{aligned} p_{i|jk} &\approx p_{i|j} \\ p_{i|jk}p_{jk}p_j &\approx p_{i|j}p_{jk}p_j \\ p_{ijk} &\approx \frac{p_{ij}p_{jk}}{p_j} \end{aligned} \quad (2)$$

More complicated pair approximations can be defined such as those based on the Kirkwood closure [23]:

$$p_{ijk} \approx \frac{p_{ij}p_{jk}p_{ik}}{p_i p_j p_k} \quad (3)$$

Higher order moment closures such as triplet approximations or  $n$ -point approximations have also been considered [32]. Likewise, in the literature of probabilistic graphical models it is common to model a complex probability distribution using factors of smaller order [6].

## 3 SPATIAL EVOLUTIONARY GAMES

We consider a population of  $M$  agents  $\{1, \dots, M\}$  that are placed on evenly-spaced points in a grid with circulatory boundary conditions. Each agent can interact with  $d$  neighbors specified by a chosen neighborhood structure. We assume that each agent interacts with their von Neumann neighborhood ( $d = 4$ ) on the grid, but the methods in this paper can easily be applied to other neighborhoods such as a Moore neighborhood ( $d = 8$ ). We will denote the set of agents in neighborhood of a given agent  $i$  as  $N(i)$ .

Let  $S$  be the set of strategies that each agent can choose from. To simplify our discussion, we will only consider memoryless strategies (excluding strategies such as Tit-for-Tat). An evolutionary game consists of  $T$  iterations, each consisting of an interaction phase followed by an update phase. Each iteration, agents update their strategies according to the rules of the evolutionary game based off of the fitness they obtain from the interaction phase.

### 3.1 Interaction phase

Each agent  $i$  chooses some action  $s_i \in S$  and receives a payoff  $\pi_i$  which is the sum of the payoffs received from playing a normal-form game with each of its neighbors. A symmetric normal form game with two strategies  $S = \{C, D\}$  has a payoff (or utility) matrix  $U$  as follows:

|   |   |   |
|---|---|---|
|   | C | D |
| C | a | b |
| D | c | d |

For this given payoff matrix, an agent playing the strategy  $C$  against a neighbor playing strategy  $D$  will receive  $b$  payoff from its interaction with that neighbor. We will denote this operation as  $U[C, D] = b$ . Since the game is symmetric, the neighbor's payoff will be  $U[D, C] = c$ . We can then write the payoff received by an agent during the interaction phase as:

$$\pi_i = \sum_{j \in N(i)} U[s_i, s_j], \quad (4)$$

where  $N(i)$  is the set of all neighbors of  $i$ .

### 3.2 Update phase

Each time the update phase occurs, a percentage of agents  $\gamma$  in the population use an update rule (see next paragraph) to decide whether to change strategies or how to reproduce on the grid. If the percentage chosen for strategy updating encompasses the entire population, then every agent will update its strategy at the same time (synchronous updating). However, in most of the EGT modeling efforts we have seen, just a single agent updates its payoff at each iteration (asynchronous updating) [12].

There are a number of different update rules ([14, Section 4.1] gives a summary). One of the best known, and the one we use in this paper, is the *Fermi rule*, in which an agent compares its payoff  $\pi$  from the interaction phase with the payoff  $\pi'$  of a randomly chosen

neighbor and switches to the neighbor's strategy with probability  $\Pr_f(\pi, \pi')$ , where

$$\Pr_f(\pi, \pi') = \frac{1}{(1 + e^{-s(\pi' - \pi)})} \quad (5)$$

where  $s > 0$  is a constant called the selection strength. Some models also define an *exploration dynamic* in which each agent can also have a small probability  $\mu > 0$  of mutation in which they change to a random strategy during the update phase [34].

## 4 BAYESIAN NETWORK APPROXIMATIONS

We start by defining a general framework for using a Bayesian Network to model the exact stochastic process present in a spatial evolutionary game. We follow this by describing a method for approximating this exact model by truncating the network. Additional steps that are analogous to moment closure are taken to complete the approximation by filling in the unknown sections of the truncated network. We will then demonstrate how these truncated networks can be used to derive existing pair approximation equations in the literature.

### 4.1 Exact Model

We first define a model that fully captures our spatial evolutionary game using a Dynamic Bayesian Network (DBN). For our example, we will model the stochastic spatial evolutionary game using the parameters listed in Table 1. The chosen game is a two strategy game with a payoff matrix that is based on the prisoner's dilemma using the synchronous ( $\gamma = 1.0$ ) Fermi-rule.

Given an evolutionary game, we define a Dynamic Bayesian Network ( $X(t), D(t), F(t)$ ), where the variable set  $X$  is split into two sets of variables  $X = A \cup \text{Pay}$  at each iteration.

- $A(t)_{i,j} \in A$ : Each of these variables represents the strategy of the agent placed at coordinate  $(i, j)$  on the grid at the start of each iteration  $t$  and its values are the strategy set  $S$ .
- $\text{Pay}(t)_{i,j} \in \text{Pay}$ : Each of these variables represent the payoff received by the agent at  $(i, j)$  during the interaction phase. The domain of these variables consists of all possible payoff values.

Each  $x_i$  is also associated with a parent function  $F(t)_i = \Pr(x_i | pa_i)$  where  $pa_i$  is the set of parent nodes of  $x_i$ . Next we will define these functions  $F(t)_i$  as Conditional Probability Tables (CPTs) for the

**Table 1: Spatial Evolutionary Game Parameters**

| Parameter          | Value   |
|--------------------|---|
| Graph Type         | Grid  |
| Graph Degree       | $d = 4$ (von Neumann neighborhood)                  |
| Update Rule        | Fermi Rule  |
| Update Percentage  | $\gamma = 1.0$                                      |
| Selection Strength | $s = 5/3$   |
| Mutation Rate      | $\mu = 0.05$  |
| Strategies         | $S = \{C, D\}$                                      |
| Payoff Matrix      | $U = \begin{pmatrix} 2 & -1 \\ 3 & 0 \end{pmatrix}$ |

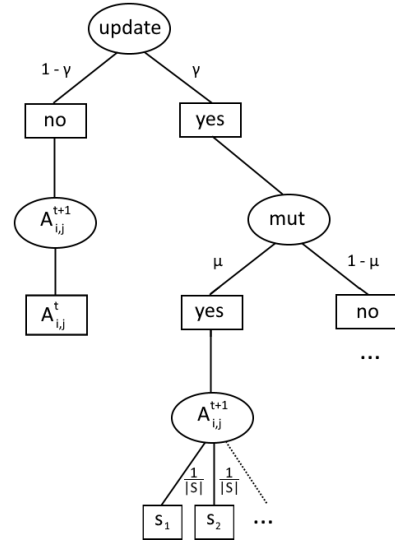
payoff nodes and the strategy nodes at time  $t + 1$ . In the following discussion we will interchange the terms "nodes" and "variables".

### 4.2 CPT for Payoff Nodes

Each node  $\text{Pay}(t)_{i,j}$  has  $d + 1$  parents:  $A(t)_{i,j}$  and its  $d$  neighbors. The conditional probability function  $P(\text{Pay}(t)_{i,j} | \text{parents})$  is constructed as a logical function using the payoff matrix  $P$ . Like in the spatial evolutionary game, we have that  $N(A(t)_{i,j})$  is the neighborhood of  $A(t)_{i,j}$ . Then we have:

$$\Pr(\text{Pay}(t)_{i,j} | A(t)_{i,j}, N(A(t)_{i,j})) = \begin{cases} 1 & \text{if } \text{Pay}(t)_{i,j} = \sum_{A(t)_{k,l} \in N(A(t)_{i,j})} U[A(t)_{i,j}, A(t)_{k,l}] \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

### 4.3 CPT for t+1 Strategy Variables



**Figure 1: Upper part of the decision tree for t+1 variables**

Each  $A(t + 1)_{i,j}$  has  $2(d + 1)$  parents:  $A(t)_{i,j}$ ,  $\text{Pay}(t)_{i,j}$  and the  $A(t)$  and  $\text{Pay}(t)$  nodes for each of the  $d$  neighbors of  $A(t)_{i,j}$ . Our goal is to define:

$$\Pr(A(t + 1)_{i,j} | \text{parents})$$

Recall that during the update phase of the evolutionary game, a percentage  $\gamma$  of agents are chosen for updating. Each agent chosen for updating chooses a random neighbor from its  $d$  neighbors to compare its payoffs with. Given a random neighbor  $A(t)_{k,l}$ ,  $A(t)_{i,j}$  has a  $\Pr_f(\text{Pay}(t)_{i,j}, \text{Pay}(t)_{k,l})$  chance of copying the strategy of  $A(t)_{k,l}$ . Additionally, separate from the Fermi rule, each agent has  $\mu$  probability of mutating to a random strategy. By conditioning on the value of each of these independent events happening, we can compute our transition probability on a case by case basis.

We follow the convention in [3] to represent this computation as a decision tree. We define the following three additional variables to represent events in the decision tree:

- update: is the node within the fraction  $\gamma$  of the population chosen for updating?

- mut: did a mutation event happen?
- rand: which neighbor did the node choose to compare its payoff with?

The decision tree can be seen in Fig. 1 and Fig. 2 where circular nodes denote variables and square nodes denote the value of the variable. The decision tree specifies the context independent paths that make up the CPT for  $A(t+1)_{i,j}$ .

In the case where a node is chosen for updating and mutation does not happen, the final probability must be conditioned on the path chosen. In Fig. 2 we have branches of the variable rand where  $A_{i,j}$  chooses different neighbors  $A_{ne^i}$  to compare its payoff to. For each neighbor  $A_{ne^i}$ , the probability of the bottom edges of Fig. 2 is:

$$\Pr(A(t+1)_{i,j} = s) = \begin{cases} \Pr_f(\text{Pay}(t)_{i,j}, \text{Pay}(t)_{ne^i}) & \text{if } A(t)_{ne^i} = s \\ 1 - \Pr_f(\text{Pay}(t)_{i,j}, \text{Pay}(t)_{ne^i}) & \text{if } A(t)_{i,j} = s \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

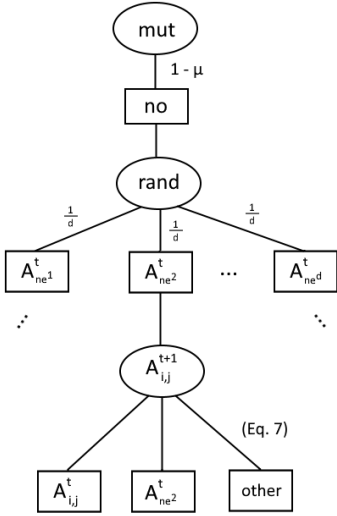


Figure 2: Cross section of (mut = no) branch

We can obtain also an explicit representation for  $\Pr(A(t+1)_{i,j} = s_{t+1} | A(t)_{i,j} = s_t, \text{other parents})$ . For notation purposes, we use indicator functions to define the following quantities:

$$\Pr_\delta = \mathbb{1}_{A(t+1)_{i,j}=A(t)_{k,l}}$$

$$\Pr_\emptyset = \mathbb{1}_{A(t+1)_{i,j} \neq A(t)_{k,l}}$$

Following a case by case breakdown, we can then write:

$$\Pr(A(t+1)_{i,j} = s_{t+1} | A(t)_{i,j} = s_t, \text{other parents}) = (1 - \gamma)\Pr_\emptyset + \gamma \left[ \frac{\mu}{|S|} + (1 - \mu) \sum_{A(t)_{k,l} \in N(A(t)_{i,j})} \frac{1}{d} \Pr_f(k, l) \Pr_\delta (1 - \Pr_\emptyset) + \Pr_\emptyset \right] \quad (8)$$

where  $\Pr_f(k, l) = \Pr_f(\text{Pay}(t)_{i,j}, \text{Pay}(t)_{k,l})$ . The final network can be seen in Figure 3, which displays a subset of the full network that contains all of the immediate parents of  $A(t+1)_{1,1}$ . Several of the parent nodes  $A(t)_{k,l}$  of the payoff nodes have been omitted in this cross-sectional view.

The resulting DBN formulation fully encodes the stochastic process of the spatial evolutionary game. For a given stochastic spatial evolutionary game, an agent-based simulation is equivalent to running a Monte-Carlo simulation on the corresponding DBN. Due to the symmetry of the grid structure, the marginal distribution of each  $A(t)_{i,j}$  for all  $(i, j)$  are the same. Therefore, one can obtain the proportion of a population playing a given strategy at time  $t$  by querying the marginal distribution of any  $A(t)_{i,j}$ . Quantities such as fixation probabilities as in [26] can also be calculated by querying the Bayesian Network for the probability that all of the strategy nodes  $A(T)_{i,j} \in A$  have the same strategy at the time horizon  $T$ .

The method for defining a DBN can be applied to other strategy update rules such as the Death-Birth or Birth-Death rules. In those cases, it would be necessary to compute different values for equation 8. For more complicated update rules such as an extended Death-Birth rule where dead nodes are not always replaced every iteration, it may be necessary to add an additional layer of nodes in the network to represent intermediate states.

It is well known that exact inference is exponential in size of the network. Therefore, if the number of agents  $M$  is large, computation will take too much time. To address this computation issue we propose a novel method for truncating the full DBN taking inspiration from moment-closure methods in the mean-field approximation literature.

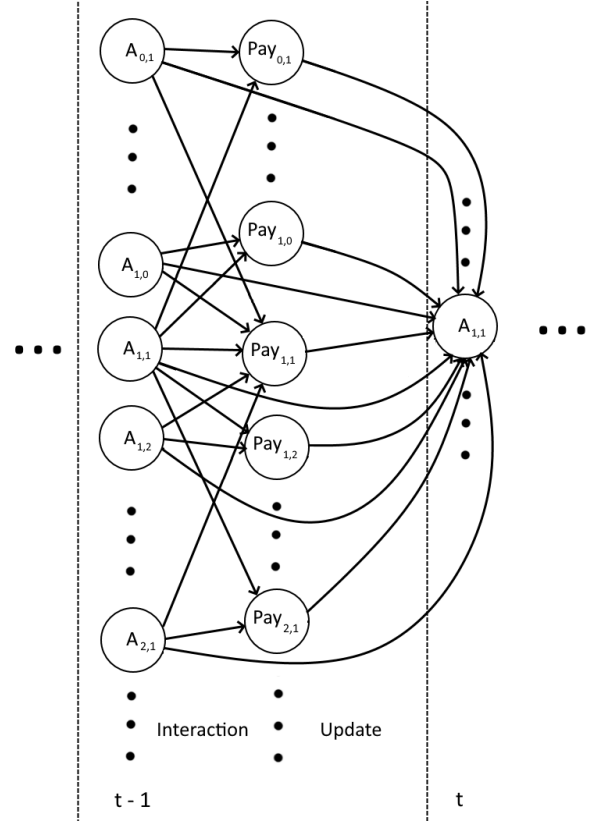


Figure 3: Slice of Dynamic Bayesian Network for the Fermi update rule centered at the agent located at position (1,1)

## 5 TRUNCATION APPROXIMATION

In our approximation method, we construct a separate Bayesian Network for each iteration that takes the states of each agent from  $t$  to  $t + 1$ . We term the nodes at time  $t$  the input nodes and the nodes at  $t + 1$  as the output nodes for the Bayesian Network at iteration  $t$ . Unlike the full DBN in the exact model, the number of input nodes at  $t$  and the number of output nodes at  $t + 1$  are not the same.

Our target is to estimate the joint probability distribution over the truncated neighborhood at time  $t$ , and use the transition probabilities as described in Figure 3 of the agent variables at time  $t + 1$ . Notice that we have the exact distribution over the truncated neighborhood at  $t = 0$  which is the product of all the single agents marginal probabilities defined by the initial condition. However, the distribution of the truncated neighborhood of subsequent iterations is no longer sparse as the strategies may have significant probabilistic dependencies which are hard to compute. Thus the idea is to approximate the joint probability of the agents in the input of any iteration using only marginal probabilities on single variables or pairwise probabilistic quantities. Because of the high symmetry between the agents, those will be derived from representative agents distributions at the output of the previous iteration.

The method consists of three steps. First, we decide on what subset of agents will make up the truncation neighborhood. The neighborhood defines what agents are represented as input nodes in the Bayesian Network at the start of each iteration. Second, we decide on what distributions to extract from the  $t + 1$  nodes that will be used to define the strategy state nodes for the next iteration. These could be marginal distributions over single output nodes, joint distributions over multiple output nodes, or conditional distributions of an output node conditioned on other nodes. Finally we decide how to define the input nodes for the network at  $t + 1$  using the distributions obtained by querying the output layer of the previous Bayesian Network.

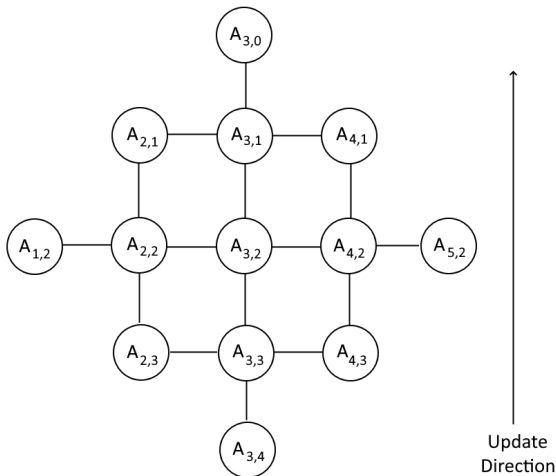


Figure 4: Truncation Neighborhood

For our example, we can exploit the symmetry of the spatial structure of the spatial evolutionary game when deciding on a truncation neighborhood. As mentioned in the previous section,

the marginal distribution of any node  $A(t)_{i,j}$  on the network is the same as the marginal distribution for every other  $A(t)_{k,j}$ . For the purpose of this truncation algorithm, we will denote arbitrarily  $A(t)_{3,3}$  as the focal node. An example truncation neighborhood is shown in in Fig. 4. This composes the set of nodes that will be the input nodes to a given Bayesian Network at each iteration.

During the update process in the spatial evolutionary game, an agent only looks in one direction to find a neighbor to update their strategy. In addition to spatial symmetry, the grid structure of our population possesses rotational symmetry. We will therefore designate one neighbor as representative for the focal agent's update. Using this rotational symmetry, we choose to look at an arbitrary update direction and assume that the node  $A(t)_{3,3}$  will update by looking at  $A(t)_{3,2}$ . We essentially truncate the branches of the random variable in Fig. 2 to have only one neighbor to choose from with a probability of 1.

By exploiting symmetry and choosing an arbitrary update direction beforehand, this particular choice of neighborhood encapsulates all possible agents at time  $t = 0$  that can affect the states of the focal node  $A_{3,3}$  and its neighboring node  $A_{3,2}$  at time  $t = 1$ . We then follow the method in the previous section to construct the Bayesian Network for  $t = 1$  and use the transition probabilities from time  $t$  to time  $t + 1$  as defined in Figure 3. Due to the limited number of nodes at time  $t$ , we are unable to construct  $t + 1$  versions of each node in the truncated neighborhood. With this truncated neighborhood, the output nodes will consist of the focal node and its neighbor  $A_{3,2}$  instead of all the nodes in the truncation neighborhood. The resulting network can be seen in Figure 5. Unlike Figure 3 which is a cross-section of the full DBN, Figure 5 displays the size of the entire Bayesian Network at  $t = 0$ . An additional node labeled "edgeUpdate" is also included that is simply the joint distribution over  $A_{3,3}(t + 1)$  and  $A_{3,2}(t + 1)$ .

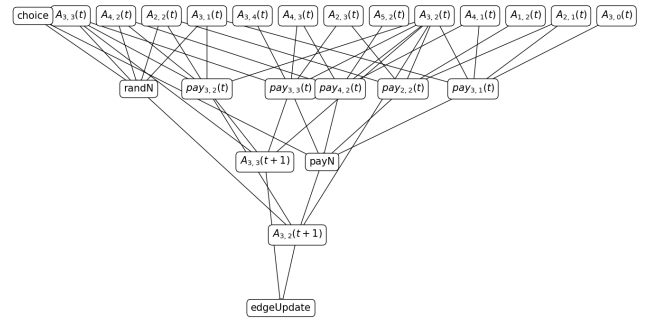


Figure 5: Truncated Bayesian Network at  $t = 0$

The resulting network is small enough that we can run exact inference algorithms such as variable elimination in a reasonable amount of time in order to compute the needed marginal and pairwise conditional distributions of the output nodes at  $t + 1$ . Clearly, in principle, we can also derive higher distributions such as the entire distribution over the 5 agent variables ( $A_{3,3}, A_{3,2}, A_{2,3}, A_{3,4}, A_{4,3}$ ). For this example, we will query the marginal distribution of the focal node and the conditional distribution of  $A_{3,2}$  conditioned on the focal node at  $t + 1$  in order to calculate the probability

distributions  $P_{s_i}, P_{s_i|s_j}$ . Since we assumed that  $A_{3,3}$  will reproduce by looking at  $A_{3,2}$ , the probability of  $A_{3,2}$  conditioned on  $A_{3,3}$  is not the same as the probability of  $A_{2,3}, A_{3,4}, A_{4,3}$  conditioned on  $A_{3,3}$ . Since we don't know the  $t + 1$  distributions of  $A_{2,3}, A_{3,4}, A_{4,3}$  conditioned on  $A_{3,3}$ , we will approximate  $P_{s_i|s_j}$  using  $P_{s_i}$  through a weighted average.

$$P_{s_i} = \Pr(A_{3,3}(t+1) = s_i), \forall s_i \in S \quad (9)$$

$$P_{s_i|s_j} = \frac{1}{d} \Pr(A_{3,2}(t+1) = s_i \mid A_{3,3}(t+1) = s_j) + \frac{d-1}{d} P_{s_i}, \forall s_i, s_j \in S \quad (10)$$

The next step in the approximation algorithm is to define the distribution of the strategy state nodes at the next iteration. At  $t = 0$ , the input nodes followed a distribution defined by the initial condition. Because the output nodes of the Bayesian Network represent fewer nodes than the input nodes, we do not have a joint distribution over the output nodes at the previous iteration to carry over to the next iteration. We approximate the joint distribution over the truncation neighborhood at the next iteration by defining the distribution of each node using only the lower order distributions stored as  $P_{s_i}$  and  $P_{s_i|s_j}$ . By doing this, we make several independence assumptions over the nodes in the input layer. This step is analogous to the idea of moment closures present in pair approximation equations.

In order to define all the nodes in the truncation neighborhood using conditional probability distributions, we need to decide on a definition order. One heuristic for this is to simply define each node in a breadth-first search manner starting from the focal node. This changes the input layer in Fig. 5 into a Bayesian network with three layers in Fig. 6.

|              |  |
|--------------|--|
| Marginal:    | $A_{3,3}$  |
| Conditional: | $A_{3,3} \rightarrow A_{3,2}, A_{2,3}, A_{3,4}, A_{4,3}$<br>$A_{3,2} \rightarrow A_{2,2}, A_{3,1}, A_{4,2}$<br>$A_{3,1} \rightarrow A_{2,1}, A_{3,0}, A_{4,1}$<br>$A_{2,2} \rightarrow A_{1,2}$<br>$A_{4,2} \rightarrow A_{5,2}$ |

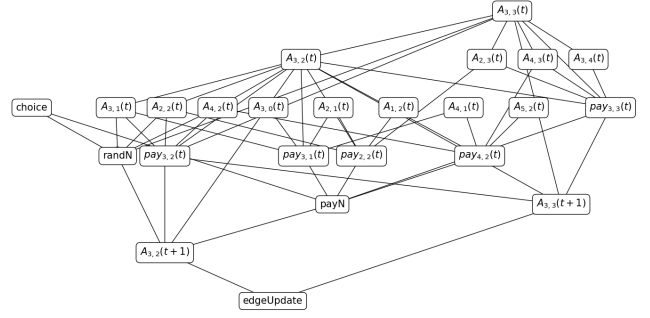
(11)

Each node is on the left hand side of an entry in the order is defined using the conditional probability distribution stored as  $P_{i,j}$ . An example is provided for parent link from  $A_{3,2}$  to  $A_{3,1}$  in Table 2. Notice that we use the conditional probabilities  $P_{i,j}$  to define nodes that are beyond the neighbors directly adjacent to the focal node. The

**Table 2: Conditional Probability Table for Node Definition**

| $A_{3,2}$ | $A_{3,1}$ | $P(A_{3,1} \mid A_{3,2})$ |
|-----------|-----------|---------------------------|
| C         | C         | $P_{C C}$                 |
| C         | D         | $P_{D C}$                 |
| D         | C         | $P_{C D}$                 |
| D         | D         | $P_{D D}$                 |

framework allows for a wide variety of truncation neighborhoods,  $t + 1$  query choices, and definition orders.



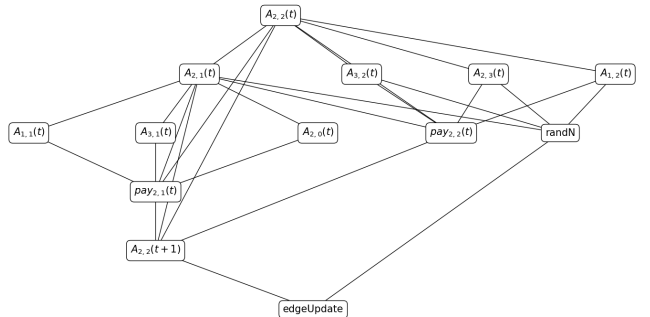
**Figure 6: Truncated Bayesian Network at  $t > 0$**

### 5.1 Pair Approximation as a Special Case

In this section, we show that there is a set of approximation parameters that result in a Bayesian Network that can be used to derive the differential equations for pair approximation. We will not provide a general proof but rather illustrate this by modeling the general spatial evolutionary game in the appendix of [12]. The strategy space consists of two strategies  $S = \{C, D\}$  and the evolutionary game uses asynchronous updating with the Fermi rule. We set our framework parameters to be:

- Truncated Neighborhood:  $\{A_{2,2}, A_{2,1}, A_3, 2, A_{2,3}, A_{1,2}, A_{1,1}, A_{3,1}, A_{2,0}\}$
- Query Values:  $P_{s_i}, P_{s_i|s_j}$
- Definition Order:

Marginal:  $A_{2,2}$   
Conditional:  $A_{2,2} \rightarrow A_{2,1}, A_{3,2}, A_{2,3}, A_{1,2}$   
 $A_{2,1} \rightarrow A_{1,1}, A_{3,1}, A_{2,0}$



**Figure 7: Truncated Bayesian Network for Pair Approximation**

We start by considering the probability that  $A_{2,2}(t+1)$  takes the value of C. We condition this on the value of  $A_{2,2}(t)$ :

$$\Pr(A_{2,2}(t+1) = C) = \sum_{s \in S} \Pr(A_{2,2}(t+1) = C \mid A_{2,2}(t) = s) \Pr(A_{2,2}(t) = s) \quad (12)$$

Recall that  $A_{2,2}(t)$  is distributed according to the current marginal distribution  $P_i$ :

$$\Pr(A_{2,2}(t+1) = C) = \sum_{s \in S} P_s \cdot \Pr(A_{2,2}(t+1) = C \mid A_{2,2}(t) = s) \quad (13)$$

Consider the probability that the focal player playing the strategy  $D$  switches to the strategy  $C$ . Let  $k_C$  be the number of neighbors of the focal node playing  $C$ . Since all nodes are independently defined, the nodes are  $C$  with probability  $P_{C|D}$ . The probability that  $k_C$  of the nodes  $A_{2,1}, A_{3,2}, A_{2,3}, A_{1,2}$  are  $C$  is:

$$\Pr = \frac{d!}{k_C!(d-k_C)!} P_{C|D}^{k_C} P_{D|D}^{d-k_C} \quad (14)$$

Without loss of generality, we assign the neighboring  $C$ -player to be  $A_{2,1}$ . We can denote  $k'_C$  as the number of nodes among the neighbors of  $A_{2,1}$  that are playing  $C$  and calculate the probability of a given configuration of  $k'_C$ .

$$\Pr = \frac{d-1!}{k'_C!(d-k'_C-1)!} P_{C|D}^{k'_C} P_{D|D}^{d-k'_C-1} \quad (15)$$

We can now condition on the nodes  $A_{2,1}, A_{3,2}, A_{2,3}, A_{1,2}$  and the nodes  $A_{1,1}, A_{3,1}, A_{2,0}$  using  $k_C$  and  $k'_C$ :

$$\sum_{k_C} \sum_{k'_C} \frac{k}{k_C} \Pr P(A_{2,2}(t+1) = C \mid A_{2,2}(t) = D, k_C, k'_C) \quad (16)$$

The value of both payoff nodes can be uniquely obtained from the amount of cooperators  $k_C, k'_C$  among the corresponding neighbors. Given that we condition on  $pay_{2,1}, pay_{2,2}, A_{2,2}$ , and  $A_{2,1}$ , the distribution of  $A_{2,2}(t+1)$  can be obtained from the the CPT of  $A_{2,2}(t+1)$ . In this case for the asynchronous Fermi rule, we have that:

$$\Pr(A_{2,2}(t+1) = C \mid A_{2,2}(t) = s, k_C, k'_C) = \begin{cases} \frac{1}{M} f(\text{Pay}_C(k'_C) - \text{Pay}_D(k_C)) & \text{if } A_{2,2}(t) = D \\ \frac{M-1}{M} + \frac{1}{M} (1 - f(\text{Pay}_D(k'_C) - \text{Pay}_C(k_C))) & \text{if } A_{2,2}(t) = C \end{cases} \quad (17)$$

Then the final probability has the form:

$$\Pr(A_{2,2}(t+1) = C) = \sum_{k_C} \sum_{k'_C} \frac{k}{k_C} \Pr(k_C) \Pr(k'_C) \left[ \frac{P_D}{M} f(\text{Pay}_C(k'_C) - \text{Pay}_D(k_C)) + P_C \left( \frac{M-1}{M} + \frac{1}{M} (1 - f(\text{Pay}_D(k'_C) - \text{Pay}_C(k_C))) \right) \right] \quad (18)$$

We can move the probability  $P_C$  out from this expression and write an equation of the form  $P_C(t+1) = P_C(t) + \Delta P_C$ :

$$P_C(t+1) = P_C(t) + \sum_{k_C} \sum_{k'_C} \frac{k}{k_C} \Pr(k_C) \Pr(k'_C) \left[ \frac{P_D}{M} f(\text{Pay}_C(k'_C) - \text{Pay}_D(k_C)) - \frac{P_C}{M} f(\text{Pay}_D(k'_C) - \text{Pay}_C(k_C)) \right] \quad (19)$$

Taking

$$\dot{p}_C = \lim_{M \rightarrow \infty} \frac{\Delta P_C}{1/M} \quad (20)$$

recovers exactly the differential equation for the time evolution of proportion of  $C$  agents in the population as in [12]. We can follow this step by step expansion of the probabilities in the Bayesian Network for the pair level probabilities and arrive at the corresponding equations for the time evolution of the pair level quantity  $p_{CC}$ .

## 6 EMPIRICAL EVALUATION

We compare our different approximation methods with agent based-simulations. The parameters for each of the approximations tested are listed in Table 4. These include the example shown in Fig. 5 and Fig. 6 called BN-Medium and a case equivalent to pair approximation called BN-PA. We also include two more cases: BN-MF and BN-Large. BN-MF has the same neighborhood as BN-PA but the input probability distribution is defined using only the marginal probabilities obtained from the previous iteration. On the other hand, BN-Large captures all the nodes that can impact a focal node directly and all their direct neighbors.

We run empirical experiments on a selection of commonly encountered games in evolutionary game theory literature (see Table 3). Our spatial evolutionary game is simulated on a 50x50 grid with parameters (with the exception of the payoff matrix) as specified in Table 1. The time evolution graphs for the simulation are obtained by averaging the results of 20 simulations on the 50x50 grid.

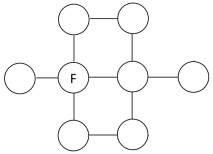
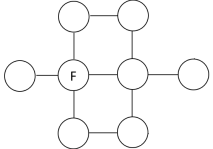
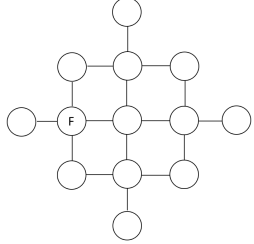
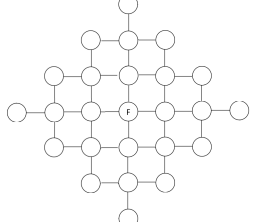
Our results show that in games where pair approximation (BN-PA) obtains a good agreement with the simulation results (Fig. 8), the larger approximation neighborhoods reduce the quantitative error in the time evolution graphs. In these cases, all approximation methods converge to the same equilibrium as the simulation. Yet, larger neighborhoods yield more accurate values for the rate of change of each strategy over time before equilibrium.

Previous research in pair approximation on 2x2 games (see [30, Section 3.8]) has indicated that pair approximation does not have good quantitative agreement with simulation results in Stag Hunt and Snowdrift games. Fig. 9 demonstrates this phenomena for a Snowdrift game, one of the pathological cases where pair approximation does not converge to the same equilibrium as the simulation. In this case we see clearly that choosing a larger neighborhood decreases the error between the approximation and the simulation. The difference between the results of pair approximation and larger neighborhood approximations can be used to indicate situations where the simulation will differ greatly from pair approximation without having to run the simulations themselves. Interestingly, on the tested games we observe that by increasing the neighborhood size we can obtain results approaching the simulation results even without using higher order probabilistic factors.

**Table 3: Evolutionary Game Payoff Matrices**

| Game Name                                  | Payoff Matrix                                   |
|--|---|
| Prisoner's Dilemma                         | $\begin{pmatrix} 2 & -1 \\ 3 & 0 \end{pmatrix}$ |
| Snowdrift                                  | $\begin{pmatrix} 2 & 1 \\ 3 & 0 \end{pmatrix}$  |
| Battle of the Sexes, symmetric version [4] | $\begin{pmatrix} 0 & 1 \\ 2 & 0 \end{pmatrix}$  |

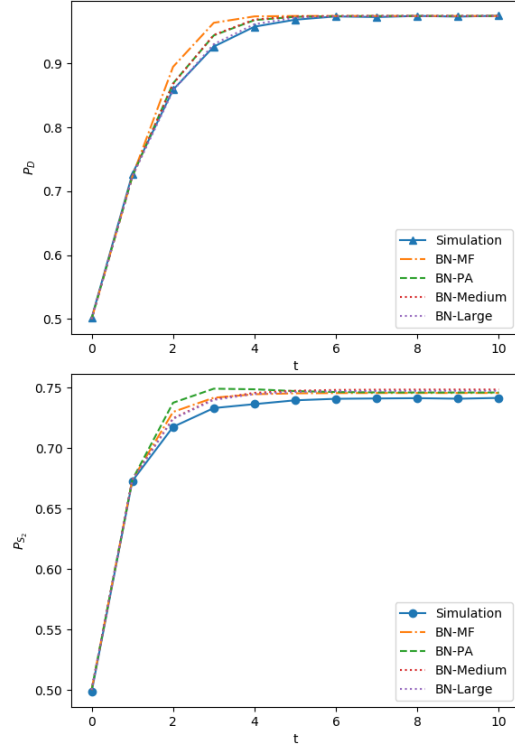
**Table 4: Approximation Framework Parameters**

| Name      | Truncation Neighborhood  | Query/Input                     |
|-----------|--|---------------------------------|
| BN-MF     |   | $P_{s_i}$ , i.i.d               |
| BN-PA     |   | $P_{s_i}$ , $P_{s_i s_j}$ , BFS |
| BN-Medium |   | $P_{s_i}$ , $P_{s_i s_j}$ , BFS |
| BN-Large  |  | $P_{s_i}$ , $P_{s_i s_j}$ , BFS |

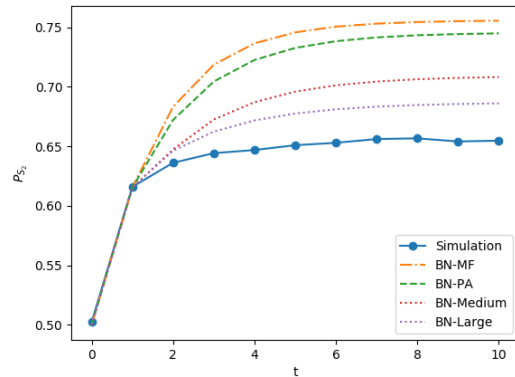
## 7 DISCUSSION AND CONCLUSION

This paper presents a dynamic Bayesian network formulation of spatial evolutionary games, thus making Bayesian network techniques applicable to such games. Since exact inference on large Bayesian networks is intractable, we accompany our formulation with a new flexible approximation scheme that is tailored to the inherent symmetry in spatial games by using a truncated neighborhood of agents that lead to truncated DBNs, which are then solved by exact inference algorithms. The truncated neighborhood can be arbitrarily defined up to the size of the population in the original stochastic simulations. By controlling the neighborhood size we can control the strength of the approximation and its complexity. We further show that certain approximated DBNs can be used to recover discrete analogs of existing pair approximations in literature. Our empirical results illustrate the potential of this approach.

Our DBN formulation is only in its initial steps. In the future, we plan to explore how to tune the approximation parameters to produce good truncated neighborhoods, and how to balance accuracy and complexity. Specifically, neighborhoods larger than BN-Medium are currently computationally intensive. However, prior work [30, 31] has shown that while pair approximations are



**Figure 8: Proportion of agents playing the second strategy in a Prisoner's Dilemma game (upper) and a symmetric Battle of the Sexes game (lower) for different approximations**



**Figure 9: Proportion of agents playing the second strategy in a Snowdrift game for different approximations**

computationally efficient, they can miss the effect of long-range correlations between agent states. We would therefore study the computation versus accuracy trade-off in a hierarchy of truncated networks. We will also explore the impact of various exact inference algorithms on such a hierarchy and the potential of using large truncated networks that are processed using approximation algorithms such as Belief propagation [6]. In particular, we will investigate what truncated network topologies are effective for approximate inference.



## REFERENCES

- [1] Christoph Adami, Jory Schossau, and Arend Hintze. 2016. Evolutionary game theory using agent-based methods. *Physics of life reviews* 19 (2016), 1–26.
- [2] Robert L Axtell. 2002. Non-cooperative dynamics of multi-agent teams. In *Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 3*. 1082–1089.
- [3] Craig Boutilier, Nir Friedmann, Moises Goldszmidt, and Daphne Koller. 2013. Context-specific independence in Bayesian networks. *arXiv preprint arXiv:1302.3562* (2013).
- [4] Russell Cooper, Douglas V DeJong, Robert Forsythe, and Thomas W Ross. 1989. Communication in the battle of the sexes game: some experimental results. *The RAND Journal of Economics* (1989), 568–587.
- [5] Ross Cressman and Yi Tao. 2014. The replicator equation and other game dynamics. *Proceedings of the National Academy of Sciences* 111, Supplement 3 (2014), 10810–10817.
- [6] Adnan Darwiche. 2009. *Modeling and reasoning with Bayesian networks*. Cambridge university press.
- [7] Soham De, Michele J Gelfand, Dana Nau, and Patrick Roos. 2015. The inevitability of ethnocentrism revisited: Ethnocentrism diminishes as mobility increases. *Scientific reports* 5, 1 (2015), 1–7.
- [8] Soham De, Dana S Nau, and Michele J Gelfand. 2017. Understanding norm change: An evolutionary game-theoretic approach. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*. 1433–1441.
- [9] Soham De, Dana S. Nau, Xinyue Pan, and Michele J. Gelfand. 2018. Tipping Points for Norm Change in Human Cultures. *Lecture Notes in Computer Science* (2018), 61–69. [https://doi.org/10.1007/978-3-319-93372-6\\_7](https://doi.org/10.1007/978-3-319-93372-6_7)
- [10] Rina Dechter. 2013. Reasoning with probabilistic and deterministic graphical models: Exact algorithms. *Synthesis Lectures on Artificial Intelligence and Machine Learning* 7, 3 (2013), 1–191.
- [11] Stephen P Ellner. 2001. Pair approximation for lattice models with multiple interaction scales. *Journal of theoretical biology* 210, 4 (2001), 435–447.
- [12] Feng Fu, Martin A Nowak, and Christoph Hauert. 2010. Invasion and expansion of cooperators in lattice populations: Prisoner’s dilemma vs. snowdrift games. *Journal of theoretical biology* 266, 3 (2010), 358–366.
- [13] Christoforos Hadjichrysanthou, Mark Broom, and Istvan Z Kiss. 2012. Approximating evolutionary dynamics on networks using a neighbourhood configuration model. *Journal of theoretical biology* 312 (2012), 13–21.
- [14] Sergiu Hart and Andreu Mas-Colell. 2001. A general class of adaptive strategies. *Journal of Economic Theory* 98, 1 (2001), 26–54.
- [15] Christoph Hauert and Michael Doebeli. 2004. Spatial structure often inhibits the evolution of cooperation in the snowdrift game. *Nature* 428, 6983 (2004), 643–646.
- [16] Christoph Hauert and György Szabó. 2005. Game theory and physics. *American Journal of Physics* 73, 5 (2005), 405–414.
- [17] Benjamin Herd, Simon Miles, Peter McBurney, and Michael Luck. 2013. Verification and validation of agent-based simulations using approximate model checking. In *International Workshop on Multi-Agent Systems and Agent-Based Simulation*. Springer, 53–70.
- [18] Josef Hofbauer, Karl Sigmund, et al. 1998. *Evolutionary games and population dynamics*. Cambridge university press.
- [19] Qing Jin, Lin Wang, Cheng-Yi Xia, and Zhen Wang. 2014. Spontaneous symmetry breaking in interdependent networked game. *Scientific reports* 4 (2014), 4095.
- [20] Christian Kuehn. 2016. Moment closure—a brief review. In *Control of self-organizing nonlinear systems*. Springer, 253–271.
- [21] Gianluca Manzo and Toby Matthews. 2014. Potentialities and limitations of agent-based simulations. *Revue française de sociologie* 55, 4 (2014), 653–688.
- [22] Javier Morales, Michael Wooldridge, Juan A Rodríguez-Aguilar, and Maite López-Sánchez. 2018. Off-line synthesis of evolutionarily stable normative systems. *Autonomous agents and multi-agent systems* 32, 5 (2018), 635–671.
- [23] Satoru Morita. 2008. Extended pair approximation of evolutionary game on complex networks. *Progress of theoretical physics* 119, 1 (2008), 29–38.
- [24] Mayuko Nakamaru, Hiroyuki Matsuda, and Yoh Iwasa. 1997. The evolution of cooperation in a lattice-structured population. *Journal of theoretical Biology* 184, 1 (1997), 65–81.
- [25] Hisashi Ohtsuki and Martin A Nowak. 2008. Evolutionary stability on graphs. *Journal of Theoretical Biology* 251, 4 (2008), 698–707.
- [26] Christopher E Overton, Mark Broom, Christoforos Hadjichrysanthou, and Kieran J Sharkey. 2019. Methods for approximating stochastic evolutionary dynamics on graphs. *Journal of Theoretical Biology* 468 (2019), 45–59.
- [27] Jorge Peña, Bin Wu, Jordi Arranz, and Arne Traulsen. 2016. Evolutionary games of multiplayer cooperation on graphs. *PLoS computational biology* 12, 8 (2016), e1005059.
- [28] Steve Phelps, Peter McBurney, and Simon Parsons. 2010. Evolutionary mechanism design: a review. *Autonomous agents and multi-agent systems* 21, 2 (2010), 237–264.
- [29] Marc Ponsen, Karl Tuyls, Michael Kaisers, and Jan Ramon. 2009. An evolutionary game-theoretic analysis of poker strategies. *Entertainment Computing* 1, 1 (2009), 39–45.
- [30] Carlos Pérez Roca. 2009. *Cooperation in evolutionary game theory: effects of time and structure*. Ph.D. Dissertation. Universidad Carlos III de Madrid.
- [31] György Szabó and Csaba Tóke. 1998. Evolutionary prisoner’s dilemma game on a square lattice. *Physical Review E* 58, 1 (1998), 69.
- [32] György Szabó and Gábor Fáth. 2007. Evolutionary games on graphs. *Physics Reports* 446, 4-6 (Jul 2007), 97–216. <https://doi.org/10.1016/j.physrep.2007.04.004>
- [33] Arne Traulsen and Christoph Hauert. 2009. Stochastic evolutionary game dynamics. *Reviews of nonlinear dynamics and complexity* 2 (2009), 25–61.
- [34] Arne Traulsen, Christoph Hauert, Hannelore De Silva, Martin A Nowak, and Karl Sigmund. 2009. Exploration dynamics in evolutionary games. *Proceedings of the National Academy of Sciences* 106, 3 (2009), 709–712.
- [35] Karl Tuyls and Simon Parsons. 2007. What evolutionary game theory tells us about multiagent learning. *Artificial Intelligence* 171, 7 (2007), 406–416.