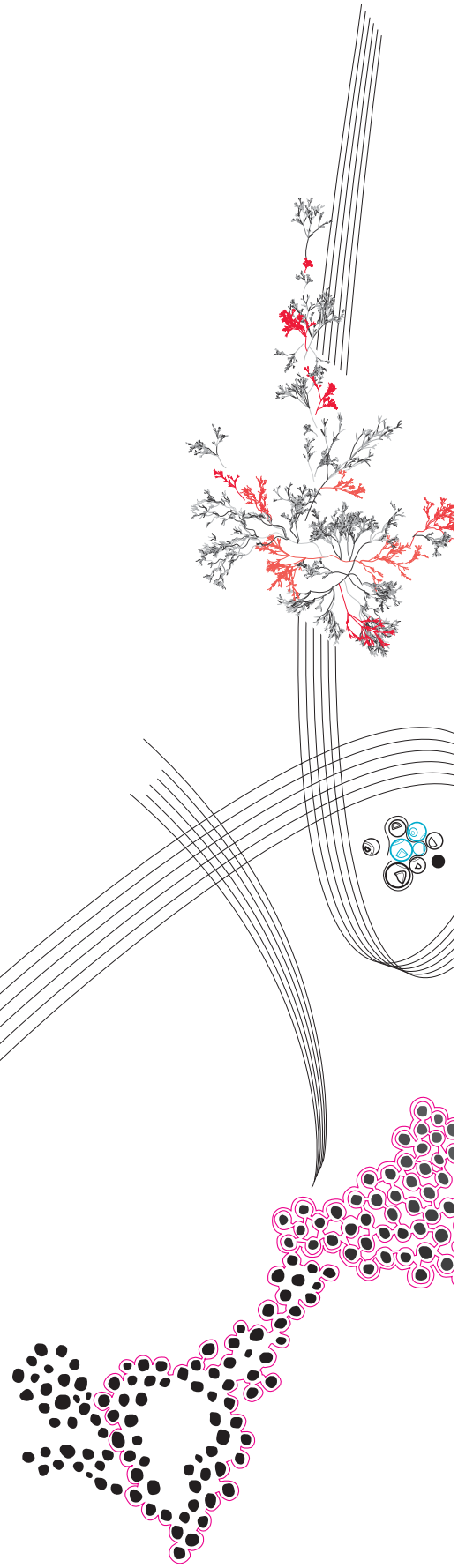


BSc Thesis Applied Mathematics and Applied Physics

Using Markov Decision theory to manipulate simple graph-based models

L.B. Meelhuijsen

Supervisor: Richard Boucherie, Maike de Jongh, Frieder Mugele

June, 2023

Department of Applied Mathematics
Faculty of Electrical Engineering,
Mathematics and Computer Science

Department of Applied Physics
Faculty of Science and Technology

UNIVERSITY OF TWENTE.

Preface

I would like to thank Richard Boucherie, Maike de Jongh, and Frieder Mugele for agreeing to be my supervisors for this assignment. I would like to especially thank Maike de Jongh for giving me valuable feedback every week. Additionally, I would like to thank my friends Thijmen Kuipers and Bram Hagens for always offering me their feedback and help, and motivating me to keep working on this assignment and study in general.

Using Markov Decision theory to manipulate simple graph-based models

L.B. Meelhuijsen

June, 2023

Abstract

We investigate the creation and destruction of opinion bubbles in small ($N = 16$) social networks. The model is constructed through a graph interpretation of the Ising model. In this model, parallels are assumed between the interaction between social individuals and information distribution and the interaction between neighbouring magnetic particles and magnetic fields. By formulating the model as a Markov Decision Process, policies for creating and destroying bubbles through field manipulation can be constructed and compared to a policy representing targeted information exposure. It is concluded that with this model, there is no significant amount of bubble creation when the individuals' opinions are perpetuated by the outside field.
Keywords: Markov Decision Process, Bubbles, Ising model, Social networks

Contents

1 Introduction

The Ising model is a model that was originally introduced to model magnetic behaviour for materials. In particular, it has been shown to be good at modelling magnetic phase transitions. This paper only concerns the 2-dimensional Ising model, which can be represented as a square lattice of points with connections only between direct horizontal and vertical neighbours. These nodes, can hold values, or spins, or orientations of $\sigma_i \in \{1, -1\}$. There is also a magnetic field applied to these nodes. The entire state, s, is then defined by the orientation of each node and the magnetic field applied to it. The Hamiltonian of the system is then of the form:

$$H(s) = -J \sum_{(i,j)} \sigma_i \sigma_j - \sum_i B_i \sigma_i, \tag{1}$$

where (i, j) represents that the two nodes are direct neighbours, B_i is the magnetic field strength on node i, and for simplicity J will be taken to be 1 throughout this paper. The system will tend towards minimizing the Hamiltonian given the magnetic field applied. This can help predict what materials do under changing conditions and how a system goes from one phase to another.

This approach has found applications in fields outside the physics of magnetism, too. Such as modelling the behaviour of cancer cells [6] and [2]. This paper will try to apply the mechanics of the Ising model to the interactions in social networks. In this application, the spins will represent the opinions or beliefs of some individuals reduced to a binary form. These beliefs are then influenced by the beliefs of the people this individual is most closely connected to, as well as the information an individual consumes through different types of media. The first is represented by the neighbour part of the Hamiltonian, and the second is represented by the magnetic field part of the Hamiltonian. Through this, the formation and destruction of bubbles or echo chambers in social networks will be investigated, first on the Ising model and then applied to the social network. The bubbles are quantified by the degree to which individuals in the system are connected to other people with the same orientation as themselves. This is represented in the first part of the Hamiltonian. This part maximizes when no individuals are connected to other people with the same orientation, and minimized when all of them are.

This is investigated by modelling the system as a Markov Decision Process (MDP). The MDP will have a state space of all the possible combinations of orientations and an action space of all the possible applied fields. This way the optimal policies can be determined for both creating and destroying bubbles. The results of these policies can then be compared to random policies and policies that represent real-world targeting of information, as well as policies based on a restricted action space. This will give insight into the mechanics of bubbles in social networks.

2 Model

2.1 MDP formulation

For the MDP formulation, it will be assumed that the Ising model will contain $N = n \cdot n$ nodes in a square lattice, where n is an even integer. Furthermore, the Metropolis algorithm will be used, meaning one randomly selected node is potentially flipped every iteration.
The state space $\mathcal{S}$ consists of all possible combinations of up and down node, meaning there are 2^N defined as $\mathcal{S} = \{\sigma \in \{-1, 1\}^{N^2}\}$.
The action-space $\mathcal{A}$ is the applied magnetic field to the model and is the same as the state space except instead of having magnitudes of 1, they have a magnitude of some constant B. Also, subsets of this action space will be formulated, namely: pure, edge, and quarter. Pure, meaning every node has the same sign. Edge, meaning only the edges can have a field applied. Quarter, meaning every quarter has the same sign. At every time step, a new magnetic field can be applied.
The reward function is described by equation 3 and is only dependent on the state and not the action.
The transition function $P(s'|s, a)$ comes from the metropolis implementation. This means that at every iteration, a random node is selected and flipped. If this flip results in a lower total energy in the system, then the flip is accepted with probability 1. Else, the flip is accepted with probability $\exp\{-\Delta E/T\}$ where T is the temperature. This implementation means that every iteration, there are N states with a non-zero transition probability, as well as a probability to stay in the same state. For any state s to a state s', where $s \neq s'$, given, an action a is:

$$P(s'|s,a) = \begin{cases} 0 & \text{if there is more than one different node in } s' \text{ compared to } s \\ 1/N & \text{if } s = s' \text{ apart from a node } (i,j), \text{ and } \Delta E < 0 \\ e^{-\Delta E/T}/N & \text{if } s = s' \text{ apart from a node } (i,j), \text{ and } \Delta E \geq 0 \end{cases}$$

With T the temperature in the system and:

$$\Delta E = -2 \cdot s_{ij}(-(s_{(i-1)j} + s_{(i+1)j} + s_{i(j-1)} + s_{i(j+1)}) - a_{ij}). \tag{2}$$

Where a_{ij} is the magnetic field at (i, j) given a, and s_{ij} is the node at (i, j).
Then:

$$P(s|s,a) = 1 - \sum_{s' \in S/\{s\}} P(s, s')$$

The reward function has two straightforward approaches. One is to minimize the number of nodes that are different to the perfect chequerboard pattern, making it a cost function strictly speaking, or maximizing its negative. That is:

$$r(s) = \min \left\{ \sum_{i=1}^{n} \sum_{j=1}^{n} s_{ij} - (-1)^{i+j}, \sum_{i=1}^{n} \sum_{j=1}^{n} s_{ij} + (-1)^{i+j} \right\}.$$

The other approach is to maximize the neighbour interaction. This is equivalent to maximizing the number of opposite neighbours. For this, the reward can simply be the first part of the Hamiltonian.

$$r(s) = \sum_{i=1}^{n} \sum_{j=1}^{n} -s_{ij}(s_{(i-1)j} + s_{(i+1)j} + s_{i(j-1)} + s_{i(j+1)}). \tag{3}$$

With an index of -1 being the same as an index of n, thus creating a taurus construction. Both of these functions maximize for perfect chequerboard patterns. However, of the two potential reward functions, Equation 3, is the more interesting one for the purpose of this paper. This is due to two reasons. First, for low-strength magnetic fields, it is unlikely for a perfect chequerboard to be created. It is, therefore, more interesting to try and minimize the bubble-like properties of the model, rather than try to reach a perfect pattern.
It was chosen to model the problem as a discounted infinite horizon problem. This is because, with an eye on the application to social networks, there is no clear finite horizon that fits the context. The value of such an MDP is defined in [5] Equation 5.4.2 to be such that for every $v^*(s) \in V$:

$$v^*(s) = \sup_{\pi \in \Pi} v_\lambda^\pi(s).$$

Where $v_\lambda^\pi(s)$ is the expected total discounted reward of a state s with a policy $\pi \in \Pi$ defined as:

$$v_\lambda^\pi(s) = \mathbf{E}\left(\sum_{t=1}^{\infty} \lambda^{t-1} r(a_t, s_t)\right),$$

with a_t being the action at time t, and s_t the state at time t. The method chosen for this implementation can be found in figure 14.4 in the book by Powell [4], and is a discounted infinite horizon value iteration method. The update step in the implementation is as follows:

$$v^n(s) = \max_{a \in \mathcal{A}} \left\{ r(s) + \gamma \sum_{s' \in \mathcal{S}} P(s'|s,a) v^{n-1}(s') \right\}.$$

We know from Theorem 6.3.1 in [5], that given the value of a state $v^*(s) \in V$ holds:

$$v^*(s) = \max_{a \in \mathcal{A}} \left\{ r(s) + \gamma \sum_{s' \in \mathcal{S}} P(s'|s,a) v^*(s'), \right\}.$$

and that for any $\epsilon > 0$, there exists an N such that for all $n \geq N$ holds:

$$|v^*(s) - v^n(s)| < \epsilon.$$

2.2 MDP implementation

Taking:

$$v^n(s) = \max_{a \in \mathcal{A}} \left\{ r(s) + \gamma \sum_{s' \in \mathcal{S}} P(s'|s,a) v^{n-1}(s') \right\},$$

the reward function can be taken out of the brackets, as it does not rely on a.

$$v^n(s) = r(s) + \max_{a \in \mathcal{A}} \left\{ \gamma \sum_{s' \in \mathcal{S}} P(s'|s,a) v^{n-1}(s') \right\}.$$

Furthermore, there is no need to sum over all $s' \in S$ as only N will have non-zero probability functions. Then there is also the fact that every state that has a non-zero transition probability, the set of these states will be called adjacent to s, $A(s)$.

$$v^n(s) = r(s) + \max_{a \in \mathcal{A}} \left\{ \gamma \sum_{s' \in \{A(s) \cup s\}} P(s'|s,a) v^{n-1}(s') \right\}.$$

Due to the way the metropolis implementation works, there is first a chance a specific node is picked to be flipped in an iteration $1/N$, then there is a probability that it flips, and goes to a different state, or stays in the current state. This lets us bring the probability to go from s to s into the summation over different states in $A(s)$ as follows:

$$v^{n+1}(s) = r(s) + \max_{a \in \mathcal{A}} \left\{ \gamma \sum_{s' \in A(s)} 1/N \left(P(s'|s,a,ij) v^{n-1}(s') + (1 - P(s'|s,a,ij)) v^{n-1}(s) \right) \right\}.$$

Where the ij in the probability function represent that the node ij in the lattice is selected to be potentially flipped during the iteration, i.e. the node where s and s' differ. Now we can use the definition of the probability function and the definition to 3 to deduce that:

$$P(s'|s,a,ij) = P(s'|s,a_{ij},ij).$$

With a_{ij} being the field strength at node ij. This can then be used to take the maximum inside the sum, as all parts of the sum are independent of each other. This means it is not necessary to find the maximum out of a set of 2^N possible actions, but rather the maximum out of 2 possible action, N times.

$$v^{n}(s) = r(s) + \frac{\gamma}{N} \sum_{s' \in A(s)} \max_{a_{ij} \in \{+B,-B\}} \left\{ P(s'|s,a_{ij},ij) v^{n-1}(s') + (1 - P(s'|s,a_{ij},ij)) v^{n-1}(s) \right\}. \tag{4}$$

To implement, the reductions to the action space are then quite straightforward. For the pure field, we take the max out of the summation again but keep the same two possibilities.

$$v^{n}(s) = r(s) + \frac{\gamma}{N} \max_{a_{ij} \in \{+B,-B\}} \left\{ \sum_{s' \in A(s)} P(s'|s,a_{ij},ij) v^{n-1}(s') + (1 - P(s'|s,a_{ij},ij)) v^{n-1}(s) \right\}. \tag{5}$$

Similarly, for the quarter implementation, but then summing over the max of each pure quarter. Lastly, for the edge implementation, it is split into two sums, once over the edges like in Equation 4, and then once over the centre nodes with the a_{ij} set to 0 for all of them. A schematic of what these restrictions on the action space look like can be found in Figure 1

2.3 Numerical implementation

The Metropolis algorithm will be used. This is important as it will influence the way the MDP is formulated and allow for a reduction in computation for the maximization of the action space, as will be discussed later. The Metropolis algorithm works by selecting a random node in the system and flipping its orientation. If the flipping of the node reduces the value of the Hamiltonian in the system, the flip will be accepted with probability 1. If the flip raises the value of the Hamiltonian, the flip will be accepted with a probability $\exp(-\Delta E/T)$, with ΔE the difference in the Hamiltonian due to the flip. This is shown in pseudocode in Algorithm 1.

The implementation creates an action lookup table for every state for a given value for B, it also gives the values of every state. This means there is a limitation to the number of

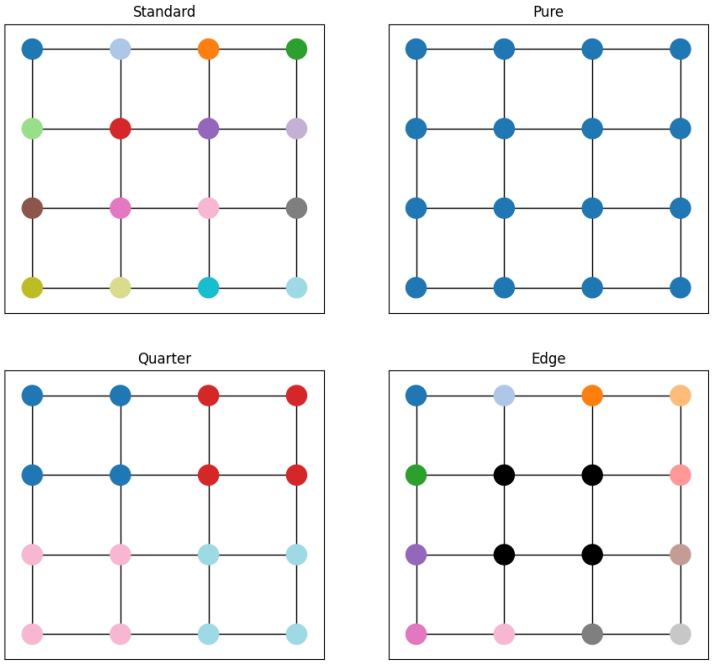


FIGURE 1: A schematic overview of what the different restrictions on the action space entail for the applied magnetic fields. In the schematic, every node with the same colour will have the same magnetic field applied, with black meaning it is always 0.

Algorithm 1 Metropolis algorithm

s = starting state
$E(s)$ = Hamiltonian
while True **do**
  for Random i, j **do**
    $s_{\text{new}} \leftarrow s$ with $s_{\text{new},ij} = -s_{ij}$
    if $E(s_{\text{new}}) \leq E(s)$ **then**
      $s \leftarrow s_{\text{new}}$
    else
      if Random.Uniform$[0, 1] \leq \exp(-(E(s_{\text{new}}) - E(s))/T)$ **then**
        $s \leftarrow s_{\text{new}}$
      end if
    end if
  end for
end while

nodes in the model, as the size of these lookup tables grows exponentially with the number of nodes. In this paper $n = 4$, is used, giving a total number of nodes N of 16. The lookup tables provide the optimal action for any possible state. The value for γ is taken to be 0.99 and the value for $\epsilon = 0.1$, where ϵ is taken to be the maximum difference overall s, between $v^n(s)$ and $v^{n+1}(s)$.
lookup tables were generated for $B \in [0, 6]$ with a step size of 0.1, for all the previously mentioned action spaces. As can be seen in Figure 2, simply scaling policies, i.e. generating a policy for one value of B and simply multiplying the actions with a constant to generate a policy for a different B, only works around the area in which the policies are scaled. Using these tables, simulations were run. These simulations consist of starting with a random initial state, the discounted node neighbour energy is calculated and added to a total. The corresponding action is extracted from the table and a metropolis iteration is performed. This is repeated for a number of iterations, in this case 2000. The discounted total node neighbour energy is then averaged over a number of instances, in this case, 1000, for every value of B for which there is a table. A shortcut used in the generation of the tables is to use the values of another B as starting values $v^0(s)$, as the difference between the two will be quite small. This reduces the number of iterations required.
The results of these simulations can then be compared between different action spaces, plotted in Figure 9, as well as against some trivial policies, plotted in Figure 6. The most trivial policy is to always apply a perfect chequerboard pattern as the action for any state. Other simple policies look only at the direct neighbours of the node selected in the metropolis iteration. The policies are characterized by the number of direct neighbours that have an opposite sign of the node selected. If the number of neighbours that has an opposite sign to the selected node is higher or equal to the characteristic number of the policy, the perfect chequerboard pattern is ignored, and instead, the action will aim to preserve the current orientation by aligning with the sign of the node. This is done for characteristic numbers $0, 1, 2, 3, 4$.

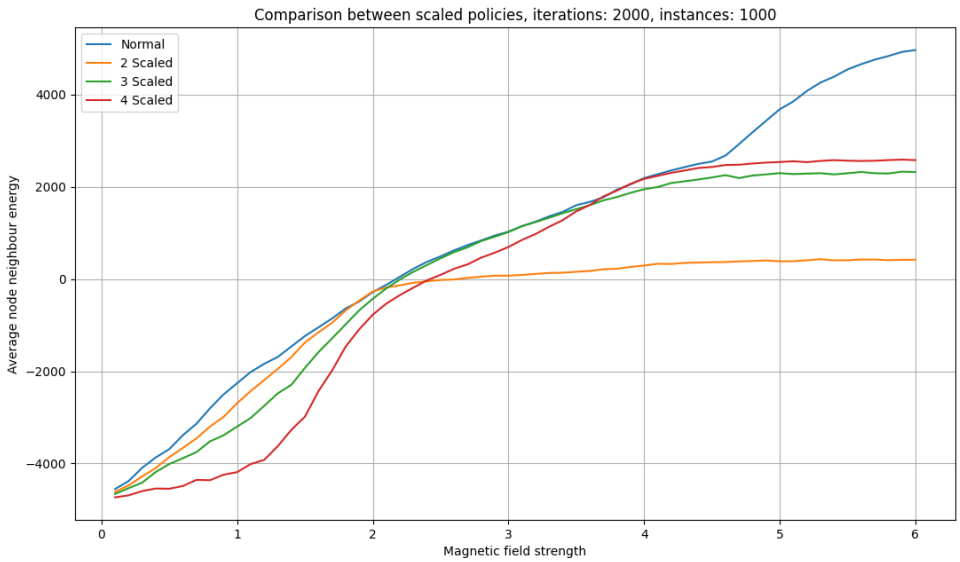


FIGURE 2: Comparison of scaled policies at different B values to taking the policy generated specifically for the B value.

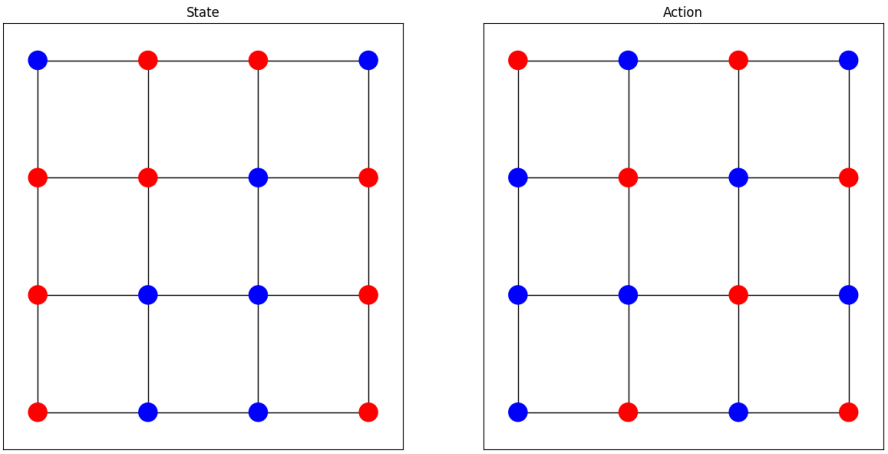


FIGURE 3: An example of a state and its corresponding best action in the standard action space with $B = 6$.

2.4 Social network implementation

The social network was modelled as an undirected graph $G(\mathbf{V}, \mathbf{E})$, where $\mathbf{V} = \{0, 1, \ldots, N-1\}$ is the set of vertices or nodes that represent the individuals in the network and $\mathbf{E}$ is the set of edges that represent a social relationship between individuals. As the Ising model can be seen as a very specific graph, the implementation can largely stay the same, the main difference being the generality of the edges. Both the state space and the (general) action space remain the same. This can be fixed by generalizing the Hamiltonian for any graph:

$$H = -J \cdot \sum_{i,j \text{ s.t. } (i,j)\in\mathbf{E}} \sigma_i \sigma_j - B \cdot \sum_{i\in\mathbf{V}} \sigma_i. \tag{6}$$

The function for the difference in energy, as in Equation 2, can then be rewritten as:

$$\Delta E = -2 \cdot \sigma_i \cdot (-a_i + \sum_{j \text{ s.t. } (i,j)\in\mathbf{E}} \sigma_j). \tag{7}$$

With a_i the action for node i. The reward function as described by Equation 3 can be rewritten as:

$$r(s) = \sum_{i\in\mathbf{V}} -\sigma_i \sum_{j \text{ s.t. } (i,j)\in\mathbf{E}} \sigma_j. \tag{8}$$

However, there will be an additional reward function introduced. A reward function that will try to maximize bubbles. Naively, this can be taken to be the negative of Equation 8:

$$r(s) = \sum_{i\in\mathbf{V}} \sigma_i \sum_{j \text{ s.t. } (i,j)\in\mathbf{E}} \sigma_j.$$

However, this would maximize when all nodes have the same orientation. Although that is still interesting, it might not capture the full extent of what it means to create bubbles in a social network. This function can be generalized by adding an additional variable:

$$d = \sum_{i\in\mathbf{V}} \sigma_i.$$

This variable is a measure of the difference in the number of nodes in each orientation. By then constructing a function, $f(d)$ the reward function can be steered towards a certain distribution:

$$r(s) = \frac{1}{f(d)} \sum_{i \in \mathbf{V}} \sigma_i \sum_{j \text{ s.t. } (i,j) \in \mathbf{E}} \sigma_j. \tag{9}$$

A simple example of such an $f(d)$ are:

$$f(d) = 1 + \left| \frac{(d-c)}{b} \right|^a. \tag{10}$$

With a, b, and c being adjustable constants to give weights to different parts of the state space, and the 1 being there to make sure there is no division by zero. These centre the weight around $d = c$. The reward function used for the construction of the lookup tables takes $a = 2$, $b = 4$, and $c = 0$, giving:

$$r(s) = \left(1 + \left| \frac{(d)}{4} \right|^2 \right)^{-1} \sum_{i \in \mathbf{V}} \sigma_i \sum_{j \text{ s.t. } (i,j) \in \mathbf{E}} \sigma_j. \tag{11}$$

There will also need to be a policy that represents an algorithm designed to give any consumer of information what they would like to see. This means a policy where:

$$a_i = B \cdot \sigma_i, \quad \forall i \in \mathbf{V}. \tag{12}$$

As well as a completely random policy to compare the effect:

$$P(a_i = B) = P(a_i = -B) = \frac{1}{2}, \quad \forall i \in \mathbf{V}. \tag{13}$$

2.4.1 Graph construction

There are a number of options for constructing a graph that represents a social network. The two that were considered here are the preferential attachment method [1] to create a 'scale-free' graph, and the 'Watts-Strogatz' model, to create a 'small-world' graph [1] [3] [9].

The preferential attachment model works by generating a graph one node at a time and adding edges between a new node and the old ones with a probability based on the degree of the old nodes. This means high-degree nodes are likely to gain more edges. This gives a power law degree distribution, which is an attribute a lot of networks have [1], and arguably social networks too. However, this mainly starts to take effect for graphs with a large number of nodes. As can be seen in [8], for the low degrees the power law can often not hold.

The alternative is to use the 'Watts-Strogatz' model. This model creates a graph with 'small-world' properties. The properties of interest are the average 'cliqueness' or 'clustering coefficient' and the mean shortest path length. The clustering coefficient c_i of a node i is defined by the ratio between the number of adjacent nodes that are adjacent to each other divided by the maximum possible amount of adjacent nodes that are adjacent to each other. Taking the set $\Omega_i = \{j \text{ s.t. } (i,j) \in \mathbf{E}\}$, the clustering coefficient is defined by:

$$c_i = \frac{\sum_{j \in \Omega_i} |\Omega_i \cap \Omega_j|}{|\Omega_i| \left(|\Omega_i| - 1 \right)}. \tag{14}$$

Then the average over all nodes can be taken as the clustering coefficient of the graph, C:

$$C = \frac{1}{N} \sum_{i \in \mathbf{V}} c_i. \tag{15}$$

The taking $l(i, j)$ to be the shortest path length between the nodes i and j, measured in amount of edges traversed, the mean shortest path length is then, as given by [3]:

$$\ell = \frac{1}{N(N-1)} \sum_{i \in \mathbf{V}} \sum_{j \neq i \in \mathbf{V}} l(i, j). \tag{16}$$

It has been observed in social networks that the mean shortest path length is always quite low as seen in [8], where is shown that in the Facebook social network essentially all users are at most 6 degrees of separation away from each other. The clustering coefficient is also investigated and is shown to be, on average, around 0.5 to 0.2 for the lower-degree nodes. These are values to keep into account when constructing the random graphs to represent social networks. It must be noted that in [3] it is proven analytically and shown numerically that the Watts-Strogatz algorithm has a mean shortest path length that grows linearly with N. This should not be a problem, however, as the networks constructed will be rather small ($N = 16$).
Qualitatively, the Watts-Strogatz algorithm takes an even positive integer $K = 2k$, a node count N and a probability β, and creates a graph with $\frac{NK}{2}$ edges. It does this by first creating a K-regular graph that can be thought of in a ring shape, with every node connected to the k nodes on both sides. It then iterates over the edges on the right side of each node and rewires them to a different node with a probability β. The node that it is rewired to is chosen uniformly, making sure to not create duplicate edges or self-loops [3] [9]. The algorithm is shown in Algorithm 2 in pseudocode.

It can be noted that the factor β acts as a measure of randomness in the graph. A

Algorithm 2 Watts-Strogatz algorithm

$V \leftarrow [0, 1, ..., N-1]$
$E \leftarrow []$
for $i \in V$ **do**
  for $j \in V$ **do**
    if $0 < |i - j| \mod (N - 1 - K/2) \leq K/2 \wedge (i, j) \notin E$ **then**
      $E \leftarrow (i, j)$
    end if
  end for
end for
for $(i, j) \in E$ **do**
  if $i < j < i + K/2 \vee (i + K/2 \geq N \wedge j \leq (i + K/2) \mod (N))$ **then**
    if Random.Uniform $[0, 1] \leq \beta$ **then**
      $(i, j) \rightarrow (i, k)$ for a random k s.t. $(i, k) \notin E$ and $k \neq i$
    end if
  end if
end for

graph built with $\beta = 0$ is completely deterministic. This randomness is not the same as would be achieved from an Eröds-Rényi random graph, as a minimum degree of k holds for every node. This randomness creates shorter paths between nodes, which is desirable,

Graph overview				
Graph	β	Expected C	Actual C	ℓ
Graph 1	0	0.5	0.5	2.4
Graph 2	0.1	0.3645	0.40625	2.175
Graph 3	0.3	0.1715	0.2815	1.983
Graph 4	0.5	0.0625	0.28125	1.9583
Graph 5	1	0	0.3125	1.9583

TABLE 1: Table with an overview of the coefficients of the constructed graphs.

as mentioned earlier. The original setup ensures that there is control over the expected clustering coefficient, as for small β this coefficient can be determined analytically to be [3]:

$$C(0) = \frac{3(k-1)}{2(2k-1)}. \tag{17}$$

With increasing β, decreasing C. This can be generalized as done in [3] for β:

$$C(p) = C(0) \cdot (1-\beta)^3. \tag{18}$$

The idea is to generate the lookup tables for a number of values of β, in the same way that was done for the Ising model. Due to the time and space, the creation of these lookup tables takes up, it was opted to make sure that the graphs picked were representative in terms of their clustering coefficient and did not have too large a mean shortest path. In doing this preliminary scanning of the produced graphs, it became clear that for the values $N = 16$ and $K = 4$ the clustering coefficients of the graphs generated were not following Equation 18. This was subsequently plotted, which can be seen in Figure 5, comparing it to graphs generated with $N = 50$. This seems to indicate that the equation does not hold very well when K and N are too close to each other, as when N was increased the clustering coefficient seems to move towards the expected values. The clustering coefficient for the ones that will be used, $N = 16$ and $K = 4$, on average seem to stay between 0.5 and 0.3, this means that the behaviour of graphs outside this range can not be investigated. However, this is within the range for the clustering coefficient previously mentioned to be representative for low degree nodes in social networks.
The values $\beta \in \{0, 0.1, 0.3, 0.5, 1\}$ were selected to have graphs constructed, and lookup tables made for both reward functions shown in Equation 8 and 11. These lookup tables were then used to simulate the behaviour of the graphs in the same way as was done for the Ising model. Additionally, the policies shown in Equations 12 and 13. An overview of the graphs constructed can be found in Table 1 and Figure 4. Some cherry-picking has been done in an effort to achieve a range of values for the clustering coefficient and the mean shortest path length. Despite this, the range is still not very broad due to the effects illustrated in Figure 5.

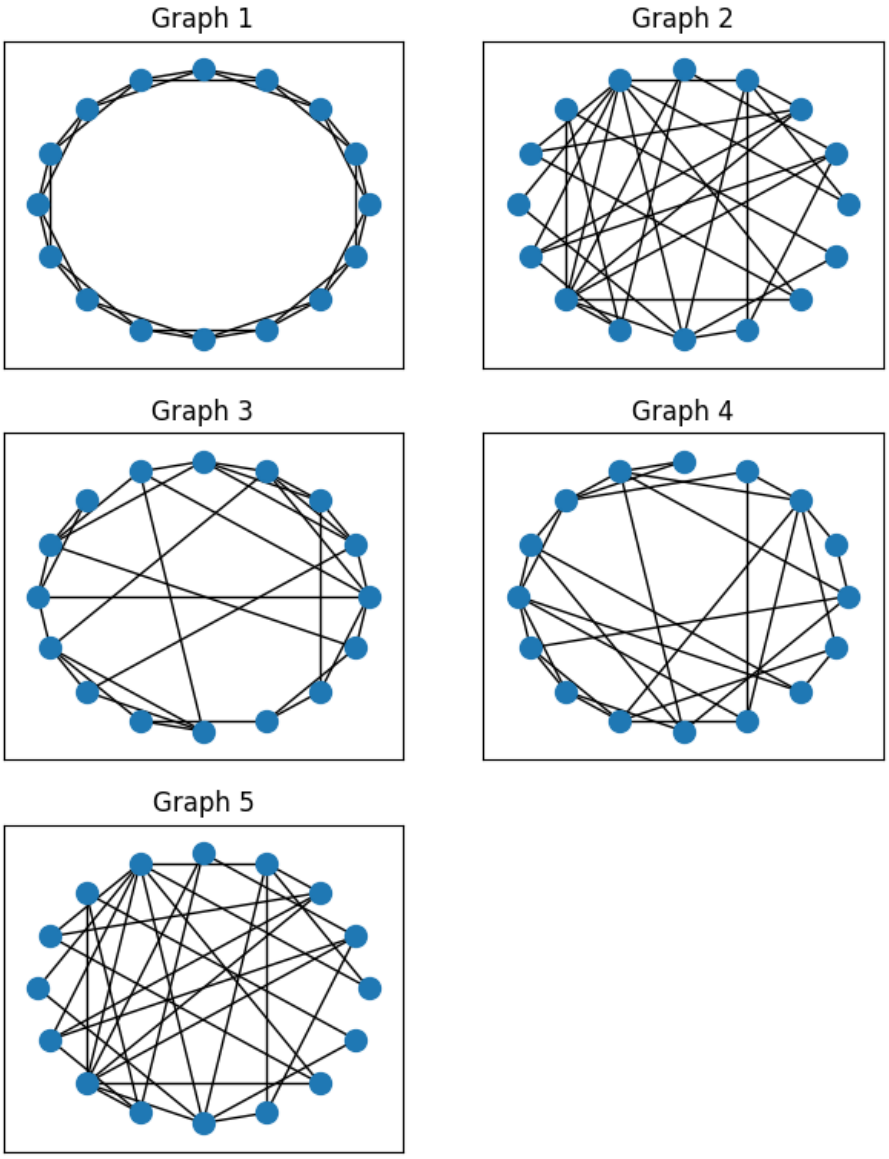
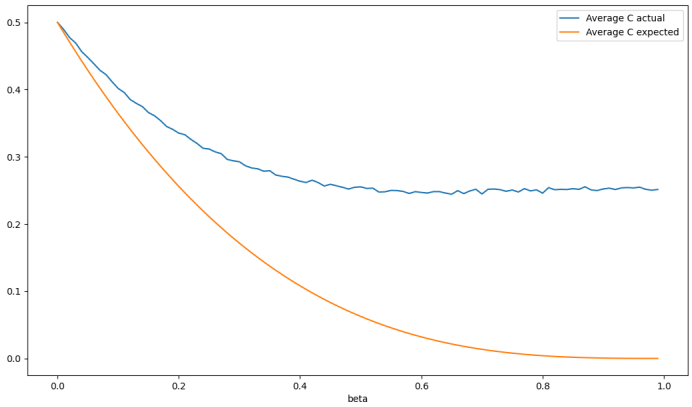
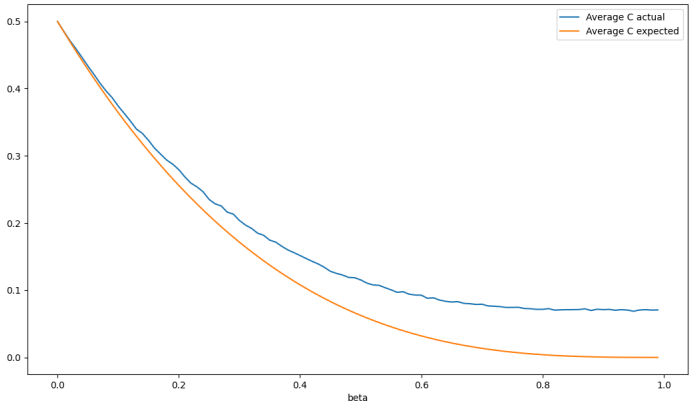


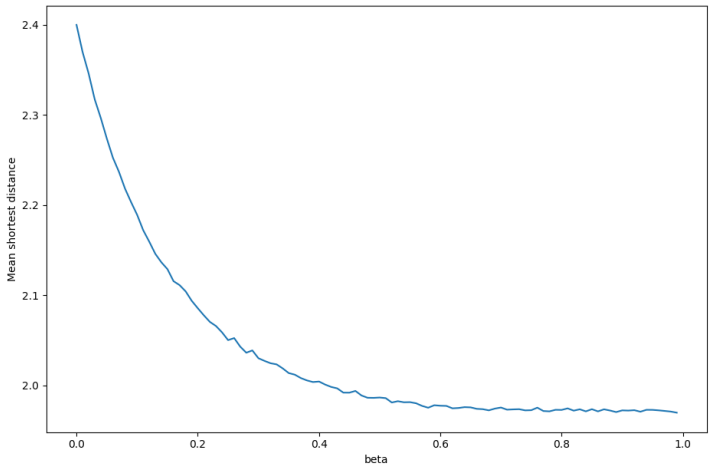
FIGURE 4: Representations of the constructed graphs for the different values of β.



(A) Average clustering coefficient for $N = 16$



(B) Average clustering coefficient for $N = 50$



(C) Mean shortest path length for $N = 16$

FIGURE 5: Comparison of the average clustering coefficient as a function of β between $N = 16$ (a), and $N = 50$ and $K = 4$ (b). As well as the expected values for ℓ for $N = 16$.

3 Results and Discussion

3.1 Ising model results

The results of the numerical experiments on the Ising model seem to indicate a couple of things: The optimal policy is dependent on the magnetic field strength. There is a significant performance difference between the optimal policy and a trivial chequerboard pattern (at least for low-field strengths). Restrictions in the action space have more significant consequences for larger magnetic field strengths.
The first observation can be made from Figure 2. The scaled policies only perform comparably to the standard method around the field strength at which the policy is constructed. when the B-values are scaled to be larger than the policy B-value, perform only marginally better than the original, which seems to indicate that the policy does not steer towards high-reward states that are hard to get to and maintain. Instead, it seems to steer towards the maximum realistically achievable state and then just tries to maintain that. This discrepancy is illustrated to a degree in Figure 7. It shows that for $B = 1$ the reward, and value of a state are less correlated than for $B = 6$, especially for the (relatively) high-value states. There seems to be a change at $B \approx 4.5$. It can be seen in Figure 6 that around this point, the optimal policy starts to perform similarly to the static chequerboard policy. The slight discrepancy between the two might be explained by the fact that there are two possible chequerboard patterns and the chequerboard policy only steers towards one of them, causing it to miss out on early rewards which are weighted more heavily than later rewards. A graph with an example of the evolution of the reward of a state throughout the iterations can be found in Figure 8. This graph is smoothened for legibility but shows clearly the increased performance with increased B-values.
The influence of a restricted action space seems to mostly matter from at $B \approx 2$ where the policy performance really starts to diverge. Beyond that, a second diverging occurs between the restricted policies at $B \approx 4$ where the edge policy start to outperform the other two significantly. At that point, the edge policy start to more reliably be able to force all edge nodes to be alternating and enables it to move to around half the total discounted reward as the unrestricted policies. The quarter policy and its strict subset, the pure policy, are unable to go much beyond a total discounted reward of 0. These policies seem to be unable to account for the randomness of the node selection. These results align with what is expected, the more restricted the policy the worse the performance, with the standard policy dominating at all B-values and the quarter policy dominating over the pure at all B-values too.

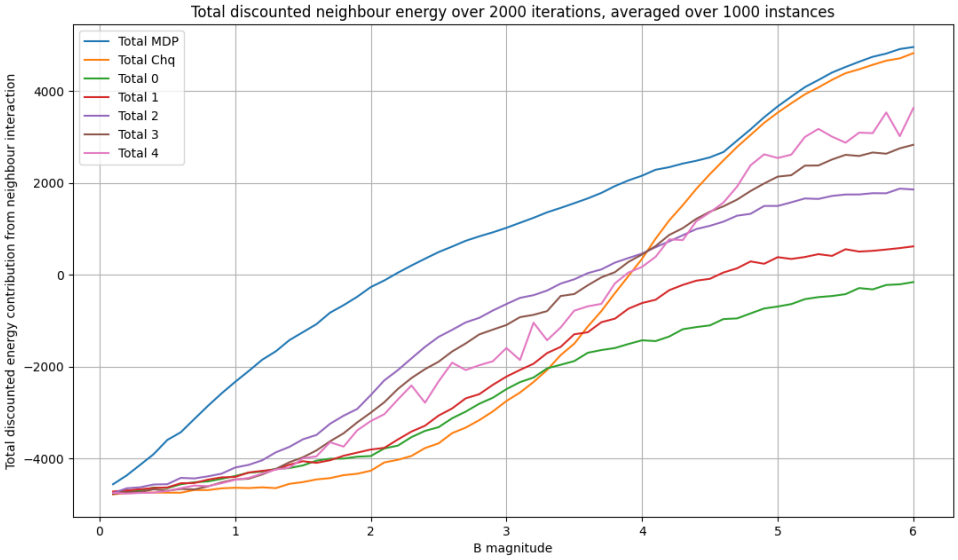


FIGURE 6: Performance of the policy from value iteration compared to some primitive policies on the Ising model. Measured by total discounted reward.

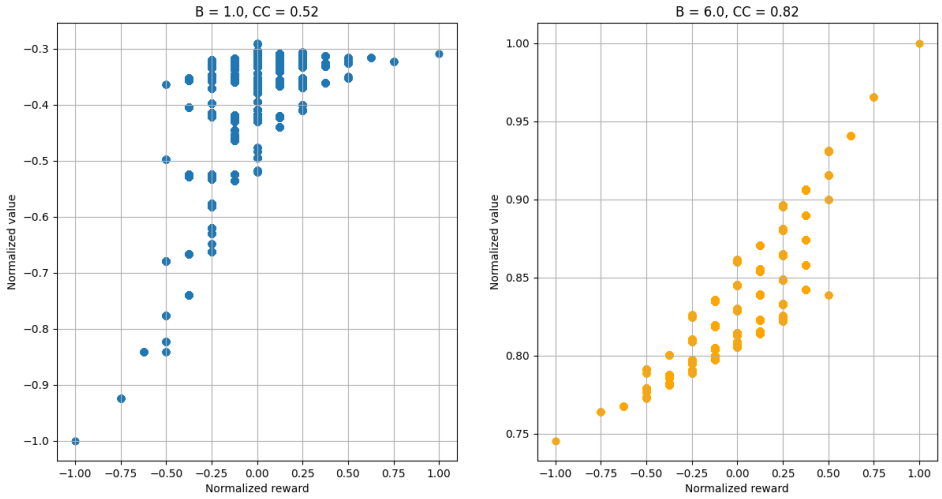


FIGURE 7: Correlation between the reward and the value for policies with B-values 1 and 6. CC is the correlation coefficient of the data.

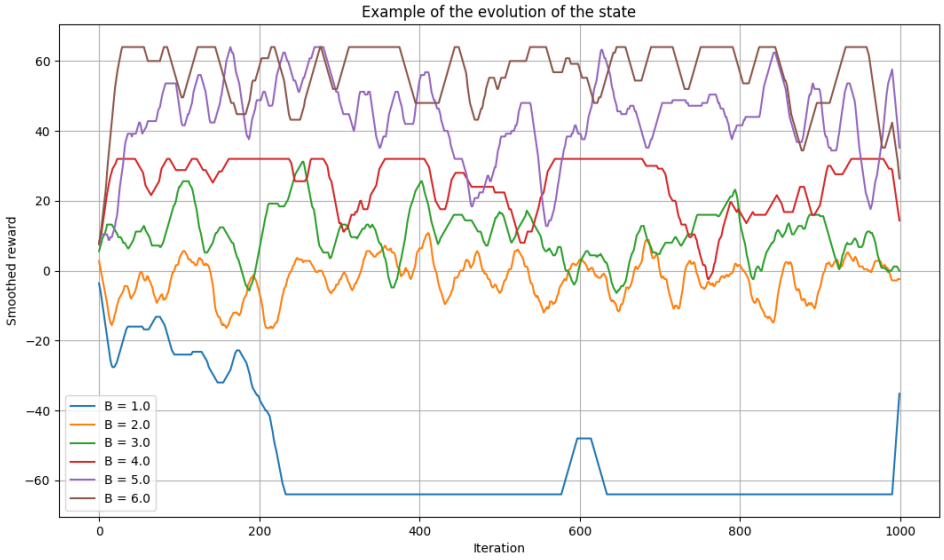


FIGURE 8: Example of how the reward of the state changes throughout the iterations for the standard Ising model at different B-values. The graph is smoothened over 20 iterations for better legibility.

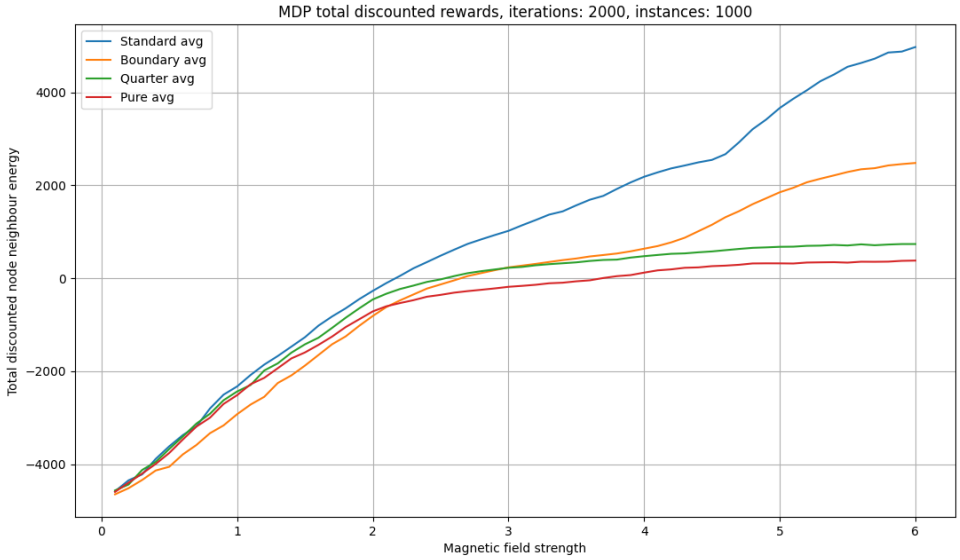


FIGURE 9: Performance of the different restrictions on the action space compared to the standard action space.

3.2 Graph implementation

3.2.1 Anti-bubble results

The results of creating lookup tables using the reward function found in Equation 8. The created lookup tables were applied to graphs given a random starting state. Then metropo-

lis iterations were performed, adding the discounted reward (using the same reward function) of the state after each iteration. This was done for 1000 iterations and then averaged over 1000 instances for every value of B.
The anti-bubble results for the social networks (Figures 10, 11, 12, 13) show some interesting behaviour that is not found in the Ising model equivalent. For the non-zero β-values, some step behaviour can be observed at the B-values 2, 3 and 4. This may be due to the variation in degrees in these graphs. Where for $\beta = 0$ and the Ising model, the degree of every node is the same and equal to 4, the nodes of the other graphs can have degrees of 2 or more (in this case the largest degree is 8 found in graph 5). At the integer values, the policies start being able to guarantee a flip from 'wrong' to 'right' for nodes of that degree. It further seems that the impact of this guarantee matters less for increasing values of B, as the step-like behaviour becomes less pronounced. This could indicate that it does not matter too much which orientation high-degree nodes have because they are likely to have neighbouring nodes with both opposite and similar orientations. This is because some neighbours are likely to be neighbours with each other as well. The policies might therefore value high-reward states where most of the nodes that are saturated with opposite orientation neighbours are also low degree, as they are more maintainable. It can also be seen that due to the fact that there are neighbours who are also neighbours with each other (something that is not possible in the Ising model), the maximum and the minimum total reward are not the same, as there are no solutions where every node has only opposite orientation neighbours.
Other than Graph 3 having a higher maximum in most cases, the non-zero β-valued graphs have very similar behaviour. Graph 3 is an outlier in neither the mean shortest path length nor the clustering coefficient, both of which are similar to that of Graph 4. Due to the small size and small sample pool of the graphs, it is possible that the difference is due to the geometry of the graph, allowing for some specific configurations that are easy to maintain and have high rewards.
It can be seen that for these graphs the quarter restriction actually performs very similarly to the edge restriction, and for graph 1 even similarly to no restriction. This could be due to the way the quartering is mapped from the Ising model to a graph. It is done by simply mapping the node number, e.g. the upper left quadrant in the Ising model would be nodes $\{1, 2, 5, 6\}$. In the Ising model, these nodes are always connected to each other, but in the graphs, this is not necessarily true. In the circular representation of Figure 4, the nodes would be numbered counterclockwise. This could allow the graphs to outperform. This effect could have been reduced by taking quadrants to be numerically increasing instead, i.e. $\{1, 2, 3, 4\}$, but even in this case, the randomness would still affect the connectedness within a quarter. Another factor that brings the quarter restriction closer is because the maximum reward for the standard version is also brought down due to the earlier mentioned neighbours of neighbours reason, and the different restrictions are consequently punished less for having non-perfect orientations.

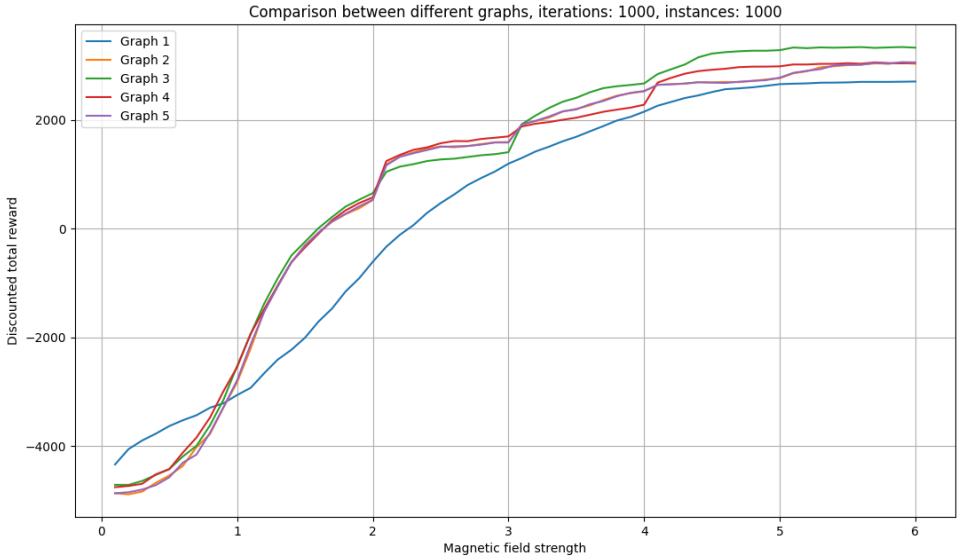


FIGURE 10: Performance of the standard action space with the discounted total reward function from Equation 8. Comparison between the different graphs.

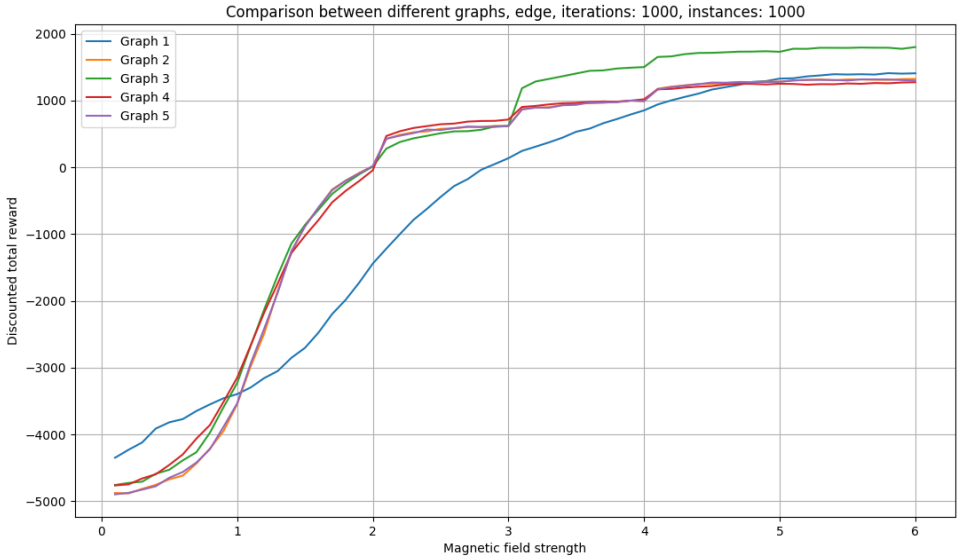


FIGURE 11: Performance of the edge action space with the discounted total reward function from Equation 8. Comparison between the different graphs.

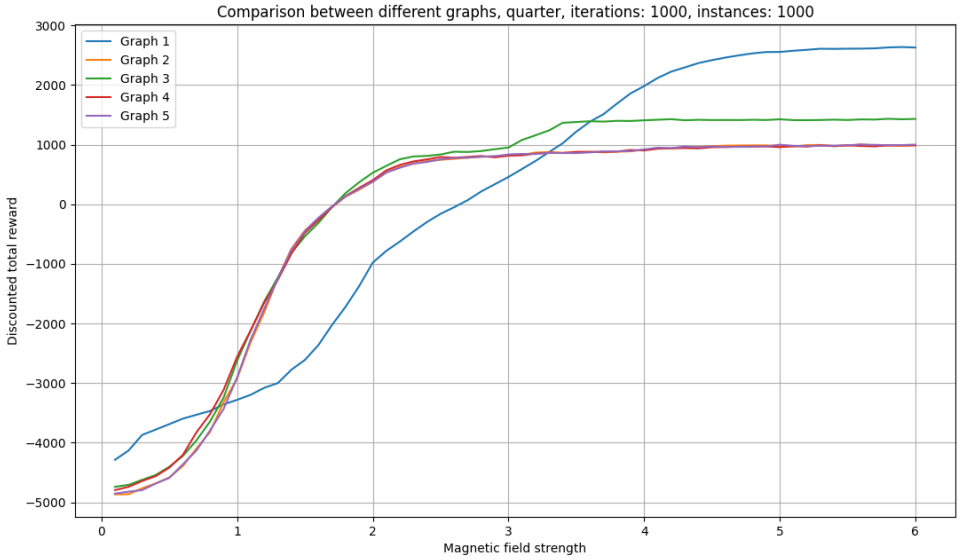


FIGURE 12: Performance of the quarter action space with the discounted total reward function from Equation 8. Comparison between the different graphs.

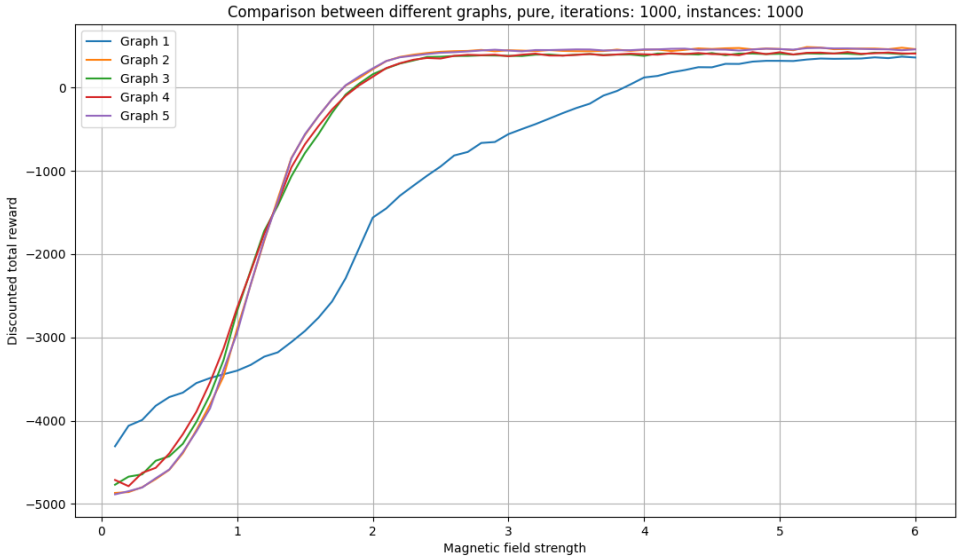


FIGURE 13: Performance of the pure action space with the discounted total reward function from Equation 8. Comparison between the different graphs.

3.2.2 Graph bubble results

The results of creating lookup tables using the reward function found in Equation 10. The created lookup tables were applied to graphs given a random starting state. Then metropolis iterations were performed, adding the discounted reward (using the same reward function) of the state after each iteration. This was done for 1000 iterations and then

averaged over 1000 instances for every value of B. The same approach was taken for the results using the simple media policy and random policy, Equations 12, 13, using the same reward function.

To start off with the results from the simple media and random policies shown in Figures 14 and 15. Other than the behaviour of Graph 1, they are very similar. They achieve their maximum between 2 and 3 with a discounted total reward between 800 and 700 and move to just below 600 to the left of that and to 0 to the right of that. The limit as $B \to 0$ shows what the total discounted reward, for the reward function in Equation 10, is when all nodes have the same orientation for most of the iterations. The reward function going to zero to the right of the maximum seems to imply that for strong fields, the average node has the same amount of opposite as similar orientation neighbours. This makes sense for both policies as the media policy largely maintains the starting state, which is random and therefore should average to zero, and the random policy steers strongly but in no particular direction. The increase in the bubble-ness from the baseline value for $B \to 0$ seems to occur when the impact of the nodes on the transition chance is similar to that of the magnetic field. This makes sense, as the node interactions will move the state towards saturation and the field, on average, to randomness, bubbles lie somewhere in between the two extremes, although it must be said that the level of bubble-ness from both policies is very low compared to the optimal policies.

The optimal policies show some strange behaviours in the Figures 16, 17, 18, and 19. The most unexpected behaviour is for the results without restrictions on the action space for the non-zero β graphs. It looks normal up to around $B \approx 1.5$ and then curves downward. Then it spikes up and down for some and then follows a more expected pattern. The fact that the optimal policies with higher values of B with no restrictions on the action space is unexpected. It must furthermore be noted that for the highest B-values, the policies only perform marginally better than the initial maximum at $B \approx 1.5$. The same behaviour can be seen in Figure 17 although less pronounced. This result could imply an implementation error. It can also imply that the assumption that the maximum allowed value for B is always optimal and only its sign matters for policies is wrong. This last explanation seems will be discussed in Section 5. Despite the dip for the intermediate values, the fact that the maximum is achieved very early, for Graph 1 as well, seems to indicate that the high reward states are quite easily maintained, which makes sense as the policies are working together with the node interactions to a larger degree than when trying to destroy bubbles.

The quarter and pure policies show slightly different behaviour in Figures 18 and 19. Apart from Graphs 1 and 2 in the quarter graph, these never recover from the initial dip. The quarter policy graph splits up with Graph 2 maintaining the same maximum, Graph 3 slightly below that and Graph 4 and 5 dipping slightly below the $B \to 0$ value and then recovering to slightly above. Graph 1 seems almost unaffected by the restricted action space. With the pure policy, not even Graph 1 maintains a value higher than the values for $B \to 0$ with large B-values. The fact that it goes below these $B \to 0$ values seems to indicate that the graphs move towards a saturated state even faster with the applied fields than without. This implies a total lack of control with this amount of restriction to the action space. This is illustrated to a degree in Figure 20. When compared to the Ising example, there is a clear lack in increased performance with higher B-values and quite violent movement.

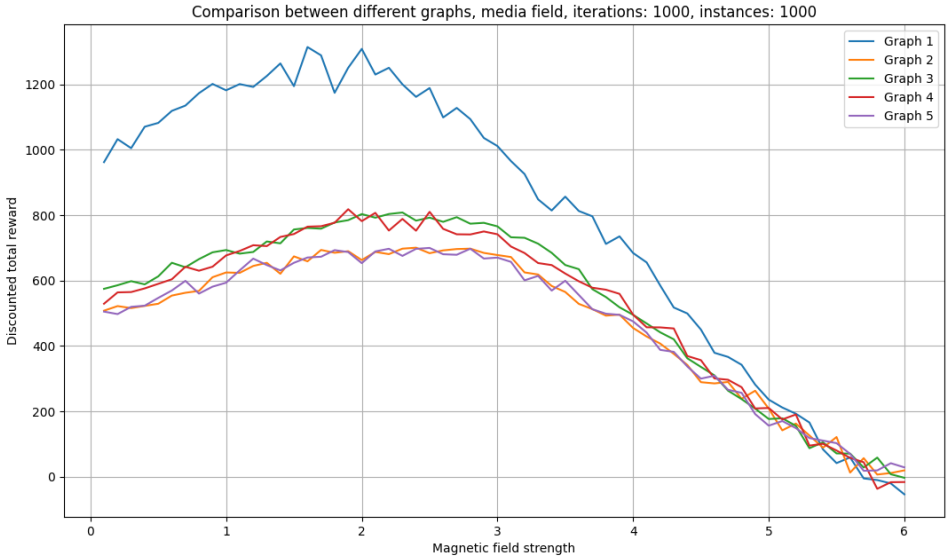


FIGURE 14: Results of applying the media policy to the different graphs, as found in Equation 12. Comparison between the different graphs with a total discounted reward function as found in Equation 10.

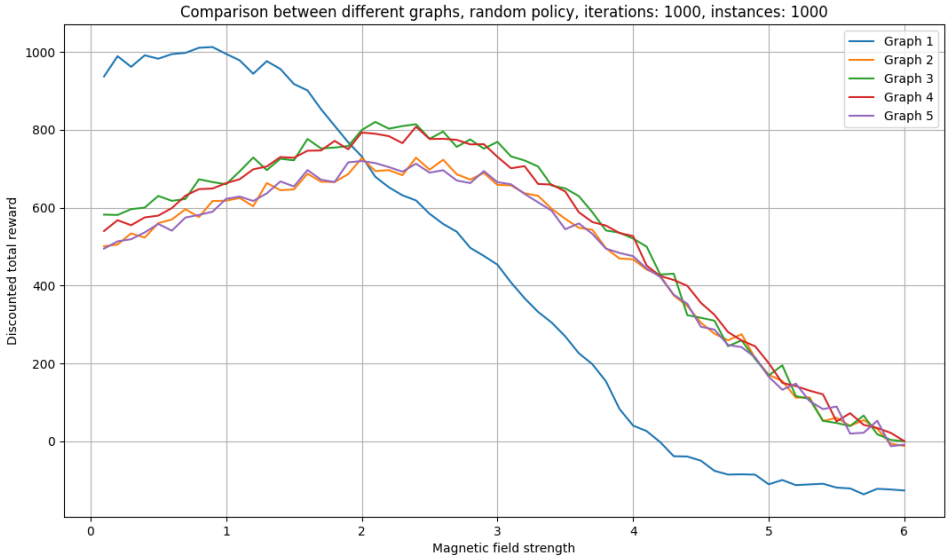


FIGURE 15: Results of applying the random policy to the different graphs, as found in Equation 13. Comparison between the different graphs with a total discounted reward function as found in Equation 10.

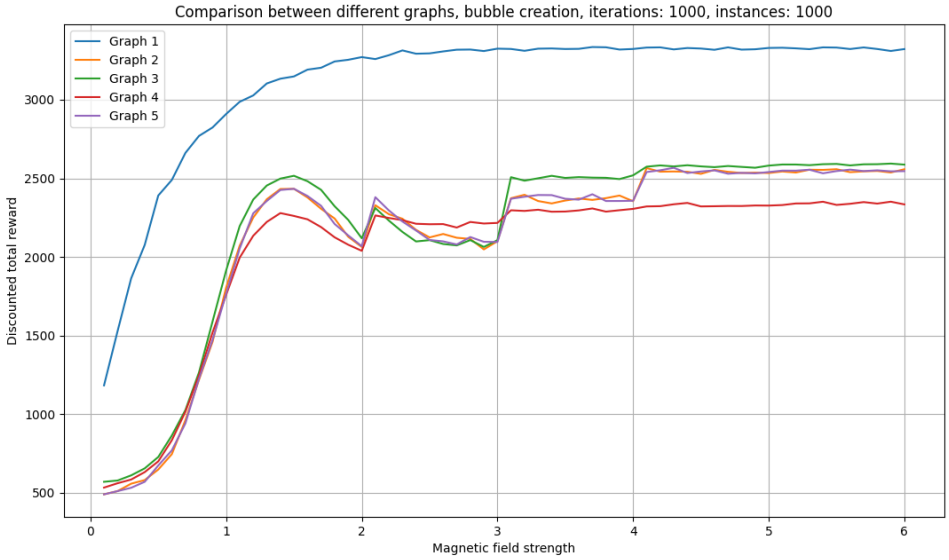


FIGURE 16: Performance of the standard action space to create bubbles, with the discounted total reward function from Equation 10. Comparison between the different graphs.

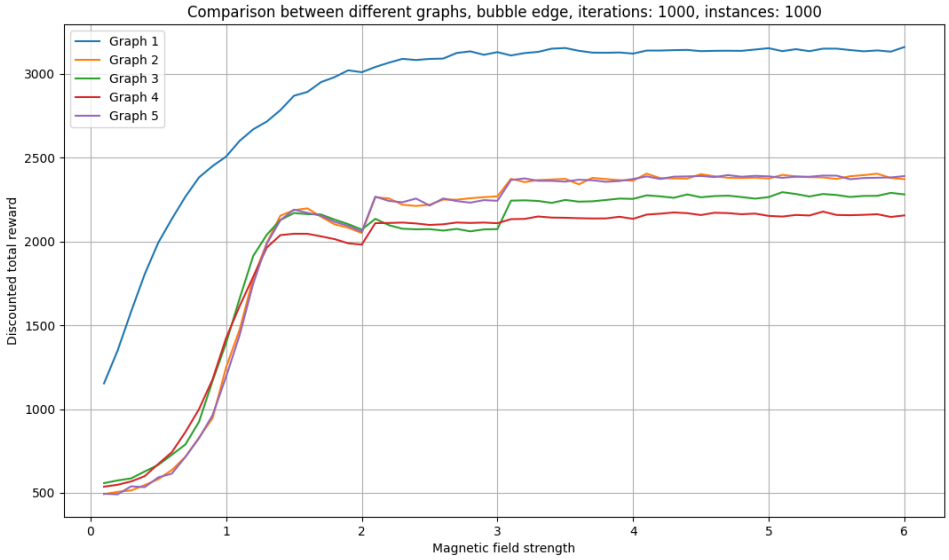


FIGURE 17: Performance of the edge action space to create bubbles, with the discounted total reward function from Equation 10. Comparison between the different graphs.

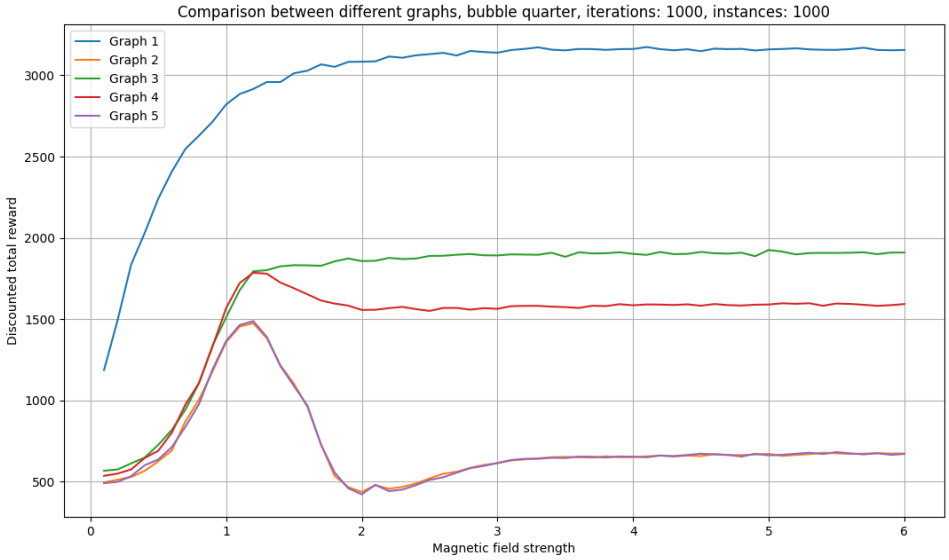


FIGURE 18: Performance of the quarter action space to create bubbles, with the discounted total reward function from Equation 10. Comparison between the different graphs.

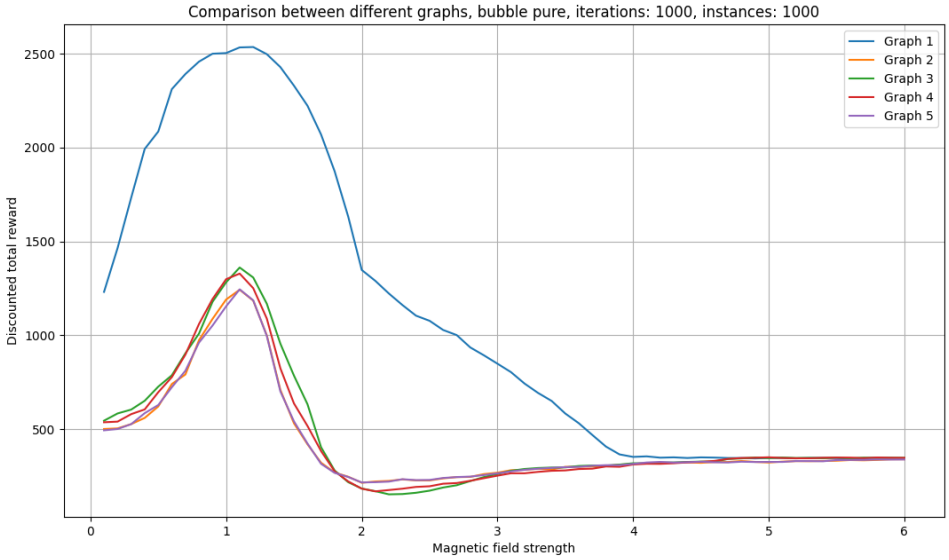


FIGURE 19: Performance of the pure action space to create bubbles, with the discounted total reward function from Equation 10. Comparison between the different graphs.

FIGURE 20: Example of how the reward of the state changes throughout the iterations for Graph 2 with the pure policy and different B-values. The graph is smoothened over 20 iterations for better legibility.

4 Conclusion

In the paper, we first explore an MDP implementation for manipulating the Ising model using applied magnetic fields. Through numerical methods, we observe that for the reward function used to maximize neighbour energy, there is a dependence of the performance of a policy on the magnetic field strength it applies. Additionally, the action space was reduced to observe its effects. This model was then reformulated to fit graphs representing social networks. Then both bubble-destroying and bubble-inducing policies were constructed for these small networks, representing social networks. The efficacy of these policies was shown through numerical simulation with random starting states. These results were compared to results generated by a media policy and a random policy. From these results, it can be concluded that using the methods described in this paper, the media policy does not generate significantly more bubbles than a random policy. Neither the random nor the media policy generate a lot of bubbles compared to the best-performing policy. It is even the case that if the action space is heavily restricted, it is impossible to generate any amount of bubble behaviour for some field strengths. The assumption that an optimal policy would always use maximum magnitude B-values allowed is not supported by the results generated by the bubble-inducing policies.

5 Recommendations

There are a few different directions that this research can be expanded upon. One is as described earlier to discard the assumption that the maximum magnitude B-value is always optimal and instead of maximizing over the set $\{-B, B\}$ in Equation 4, it is necessary to maximize over the range $[-B, B]$. This becomes quite elaborate to do analytically, as the transition probabilities are piecewise functions, and can become computationally expensive

to approximate numerically.
To really open this concept up to more interesting avenues of research, it would require a way to approximate the optimal policy in a state in real-time during the simulation stage. This can open the door to testing out a wider range of networks, as with the current approach new lookup tables must be generated and saved for each combination of graph, reward, and B-value. This severely limits the scope of the research and makes it impossible to make conclusions about the influence of the clustering coefficient and the mean shortest value. A good approximation algorithm could allow for testing with bigger and more diverse networks, and a wider range of reward functions. This last one is valuable, especially for the bubble-inducing reward function, as the current parameters are quite arbitrary.
By being able to generate policies in real time for an arbitrary graph, it would allow for a non-static network. By rewiring edges similarly to the flipping of orientations in a metropolis iteration, the dynamic nature of social networks can be modelled. In [7] it is argued that the rewiring of connections is vital for reproducing empirical data on the dynamics of social behaviour in networks. The rewiring could be approached through the same Hamiltonian as used for the nodes. During an iteration, an edge is rewired and if it reduces the Hamiltonian, the rewiring is accepted, else there is some chance of it being accepted. With such an approach to modelling the dynamic nature of social networks, the media policy will likely induce more bubbles than the random policy for high B-values. This is because the media policy will try to force the nodes to keep the same orientation and over time the Hamiltonian will minimize by rewiring the edges to only have nodes of the same orientation on both sides. The dynamics induced by introducing the rewiring this way would of course need to be corroborated by the existing literature on the dynamics of social networks.

References

[1] Réka Albert and Albert-László Barabási. "Statistical mechanics of complex networks". In: *Reviews of Modern Physics* 74.1 (Jan. 30, 2002). Publisher: American Physical Society, pp. 47–97. DOI: 10.1103/RevModPhys.74.47. URL: https://link.aps.org/doi/10.1103/RevModPhys.74.47 (visited on 06/12/2023).

[2] Didier Barradas-Bautista et al. "Cancer growth and metastasis as a metaphor of Go gaming: An Ising model approach". In: *PLOS ONE* 13.5 (May 2, 2018). Publisher: Public Library of Science, e0195654. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0195654. URL: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0195654 (visited on 03/19/2023).

[3] A. Barrat and M. Weigt. "On the properties of small-world network models". In: *The European Physical Journal B - Condensed Matter and Complex Systems* 13.3 (Feb. 1, 2000), pp. 547–560. ISSN: 1434-6036. DOI: 10.1007/s100510050067. URL: https://doi.org/10.1007/s100510050067 (visited on 06/12/2023).

[4] Igor Halperin. "Reinforcement Learning and Stochastic Optimization: A Unified Framework for Sequential Decisions: by Warren B. Powell (ed.), Wiley (2022). Hardback. ISBN 9781119815051." In: *Quantitative Finance* 22.12 (Dec. 2, 2022), pp. 2151–2154. ISSN: 1469-7688, 1469-7696. DOI: 10.1080/14697688.2022.2135456. URL: https://www.tandfonline.com/doi/full/10.1080/14697688.2022.2135456 (visited on 05/11/2023).

[5] Martin L. Puterman. "Markov Decision Processes: Discrete Stochastic Dynamic Programming". In: Wiley Series in Probability and Statistics. Edition: 1. Wiley, Apr. 15, 1994. ISBN: 978-0-471-61977-2 978-0-470-31688-7. DOI: 10.1002/9780470316887. URL: https://onlinelibrary.wiley.com/doi/book/10.1002/9780470316887 (visited on 03/18/2023).

[6] Alfonso Rojas-Domínguez et al. "Modeling cancer immunoediting in tumor microenvironment with system characterization through the ising-model Hamiltonian". In: *BMC Bioinformatics* 23.1 (Dec. 2022). Number: 1 Publisher: BioMed Central, pp. 1–25. ISSN: 1471-2105. DOI: 10.1186/s12859-022-04731-w. URL: https://bmcbioinformatics-biomedcentral-com.ezproxy2.utwente.nl/articles/10.1186/s12859-022-04731-w (visited on 03/17/2023).

[7] Kazutoshi Sasahara et al. "Social influence and unfollowing accelerate the emergence of echo chambers". In: *Journal of Computational Social Science* 4.1 (May 1, 2021), pp. 381–402. ISSN: 2432-2725. DOI: 10.1007/s42001-020-00084-7. URL: https://doi.org/10.1007/s42001-020-00084-7 (visited on 06/26/2023).

[8] Johan Ugander et al. *The Anatomy of the Facebook Social Graph*. 2011. arXiv: 1111.4503 [cs.SI].

[9] Duncan J. Watts and Steven H. Strogatz. "Collective dynamics of 'small-world' networks". In: *Nature* 393.6684 (June 1998). Number: 6684 Publisher: Nature Publishing Group, pp. 440–442. ISSN: 1476-4687. DOI: 10.1038/30918. URL: https://www.nature.com/articles/30918 (visited on 06/12/2023).