ECOLE
POLYTECHNIQUE
DE BRUXELLES

ULB

UNIVERSITÉ LIBRE DE BRUXELLES

Swarm resiliency against non-cooperative agents in collective decision-making

A computational study on different networks

Mémoire présenté en vue de l'obtention du diplôme
d'Ingénieur Civil en informatique à finalité spécialisée

Antoine Bertoux

Directeur
Professeur Marco Dorigo

Co-Promoteur
Dr. Andreagiovanni Reina

Service
IRIDIA

Année académique
2021 - 2022

French abstract

Ce mémoire étudie l'impact des agents hostiles dans la prise de décision collective. Nous avons étudié des essaims composés d'agents communiquant entre eux afin de saccordé sur la meilleure opinion parmi n alternatives (best of n problem). Nous considérons des scénarios dans lesquels les agents sont statiques et communiquent avec leurs voisins, la liste de ces voisins sont définis par des réseaux aléatoires dont la topologie est prédéfinie, ainsi que des scénarios dans lesquels les agents se déplacent sur un plan 2D et interagissent avec les agents proches d'eux (voisins). Les agents utilisent différentes stratégies de prise de décision pour traiter les informations sociales et environnementales et mettre à jour leur opinion. Nous considérons quatre stratégies de prise de décision obtenues en combinant deux mécanismes différents issus de modèles de vote de la dynamique des opinions. Le premier type de mécanismes définit comment traiter l'information sociale et peut être le "voter model rule" (VMR) ou le "local majority rule" (LMR). Le second type de mécanismes définit comment un agent change d'opinion lorsqu'il reçoit de nouvelles informations, il peut s'agir du modèle "direct switch" ou du modèle "cross-inhibition". Nous avons d'abord testé la performance des quatre stratégies dans un scénario entièrement coopératif où nous étudions l'impact de divers paramètres, tels que le nombre d'options et la topologie du réseau de communication. Ensuite, nous étudions l'impact des agents hostiles et la résilience de chaque stratégie face à eux. Nous considérons trois types d'agents hostiles : les agents fous qui choisissent une opinion aléatoire à chaque pas de temps, les zélotes qui ne changent jamais d'opinion, et les contrariens qui adoptent toujours l'opinion la moins populaire parmi leurs voisins. Notre analyse montre que certaines stratégies sont plus résilientes que d'autres et grâce à notre expérimentation modulaire, nous pouvons comprendre quel mécanisme impacte la résilience. Cependant, notre analyse montre également qu'il n'existe pas de stratégie miracle, supérieure à toutes les autres, mais qu'en fonction du scénario et du type d'agents hostiles, il existe des compromis entre les stratégies et leurs performances.

English abstract

This thesis is a broad study on the impact of hostile agents in collective decision-making. We study swarms composed of agents that communicate with each other in order to agree on the best opinion among n alternative (best of n problem). We consider scenarios in which agents are static and communicate with neighbours defined by random networks that have a predefined topology, as well as scenarios in which agents move on a 2D plane and interact with neighbours in close proximity. The agents use different decision-making strategies to process the social and environmental information and update their opinion. We consider four decision-making strategies which are obtained by combining two different mechanisms from voting models of opinion dynamics. The first type of mechanism defines how to handle social information and can be the voter model rule (VMR) or the local majority rule (LMR). The second type of mechanism defines how an agent change its opinion when it gets new information, it can either be the direct switch model or the cross-inhibition model. Our computational analysis first tests the performance of the four strategies in a fully cooperative scenario where we investigate the impact of various parameters, such as the number or options and the communication network topology. After, we study the impact of hostile agents and the resiliency of each strategy against them. We consider three types of hostile agents: the mad agents who choose a random opinion at each timesteps, the zealots who never change their opinion, and the contrarians who always adopt the least popular opinion among their neighbours. Our analysis shows that some strategies are more resilient than others and through our modular behaviour we can understand which mechanism enables such level of resiliency. However, our analysis also shows that there is no single strategy that is superior to all others but depending on the scenario and the type of hostile agents, there are trade-offs among strategies and their performances.

Acknowledgments

I would like to thanks Andreagiovanni Reina for supervising me during this master thesis and Federico Pagnozzi for helping me with the High Performance Computer (HPC) of the IRIDIA Laboratory.

Contents

List of Figures

Chapter 1

Introduction

In nature, groups of individuals often need to make a collective decision between different options. In human societies, the voting process is also a central part of the organisation of democratic countries. Enabling robot swarms to make collective decisions has interesting potential applications, for example decentralised robot swarms can accomplish tasks such as prospecting for minerals or building houses, and those tasks require the swarm to reach a consensus on one of the options (e.g., where to dig the mine or lay the fundations of new buildings). Decision-making is thus a classical bio-inspired application of swarm robotics [Valentini et al., 2017]. Being also directly linked to the behaviours of groups of human, it is a classical subject in social science [Castellano et al., 2009]. However, swarms (and human societies) may contain hostile individuals whose objective is to slow down the swarm's consensus or to trick the swarm into choosing a suboptimal solution. Moreover, applications in difficult environments may require the swarm to be resilient against broken robots which broadcast random or erroneous information. The study of the impact of hostile agents in social science is old [Mobilia, 2003] [Galam, 2004] [Galam and Jacobs, 2007], but this subject as not been studied as extensively in swarm robotics.
In this thesis, we focused on the impact of hostile agents on the decision-making process of a group of agents. For this, we have compared the behaviour of four decision-making strategies on two problems. The first is the classical best-of-n problem, where agents must select the best among n options. The second is the symmetry breaking problem, where agents must select one between two options with identical quality. The difficulty of those problems comes from the noisy environmental readings that agents make to estimate the options' qualities. We tested those four strategies by doing computational experiments (using agent based modelling) both in a fully cooperative environment and in a non cooperative environments where hostile agents are present. We conducted those computational experiments on swarm (mobile agents represented as body-less particles as in [Valentini et al., 2014]) and on static networks generated using well known models (random geometric graph, Barabasi-Albert [Barabasi and Albert, 1999] and Erdos–Rényi [Gilbert, 1959]). The four decision-making strategies are obtained by combining different mechanisms from voting models of opinion dynamics. These mechanisms define how to handle information and can be distinguished in two types of mechanisms. The first type of mechanism indicates how to process the social information coming from other agents. Two well known models exist: the called local majority rule (LMR) [Galam, S., 2002] and the voter model rule (VMR) [Clifford and Sudbury, 1973]. The mechanisms of the second type indicates how the agent can update its opinion when it gets new information. This information can either be environmental or social. This second group of models is made of the classical direct switch model (the agent switch to the new opinion) [Redner, 2019] and of the more recent cross-inhibition model [Seeley et al., 2012] (see section 3.2). An important addition to the opinion dynamics strategies studied in social sciences is that every agent estimates the quality of its opinion and its probability of broadcasting its opinion depends on the quality estimation [Valentini et al., 2014].

We used three types of hostile agents to study how those strategies would perform in a non-cooperative scenarios. The first type of hostile agent is the zealot, a zealot is an agent that never changes its opinion and always shares it with all its neighbours (it gives its opinion quality estimate the maximum value). The second type of hostile agent is the mad agents that chooses a random opinion at each step and gives it a random quality estimation. The third type of hostile agent is the contrarians, it behaves like a normal agent but when it processes the information coming from other agents, it takes the least popular (minority) opinion. All those tests were done while varying different parameters, mainly the environmental noise, the graph's average degree, the number of options in the best-of-n problem and the number of hostile agents. All the experiments were done using a multiagent simulator called DeMaMAS [University of Sheffield, 2020], to match our needs we also expanded and optimized DeMaMAS (see chapter 4).

1.1 Thesis layout

This thesis is splits into 6 chapters. The chapter 2 provides a summary of the state of the arts of the fields of decision-making and of swarm robotics, focusing on the models used in this thesis and on the impact of hostile agents. In chapter 3 we provide a clear definition of the simulator's structure and explain the different models used. Then, in chapter 4 we explain in more details the simulator's capabilities and the parameters used to configured it. We conclude by giving some information on the cluster used to run the simulations. In chapter 5 we present and give a detailed analysis of the results. It starts by the results of the experiments in a fully cooperative environment on the best-of-2, symmetry breaking and best-of-n problems. Then, the results of the experiments on a non-cooperative environment are presented both on the best-of-2 and symmetry breaking problems. Finally, the last chapter concludes the thesis by a higher level discussion of the results and some suggestions for future works.

1.2 Original Contribution

While the presented experiments have been performed using an existing simulator called DeMaMas (see section 4.1), during this thesis we improved and extended the simulator's functionalities. In particular, the original work of this thesis includes:

- Optimize the simulator (see section 4.2).
- Add a parameter to DeMaMAS that allow to stop the simulation not when the quorum has been reached, but when it has been reached and maintained for a defined number of steps (see section 4.3).
- Add to DeMaMAS the possibility of run tests on three types of graphs: random geometric, Barabasi-Albert [Barabasi and Albert, 1999] and Erdos–Rényi [Gilbert, 1959] (see section 4.3).
- Run hundreds of simulation on a SLURM base clusters using bash scripts.
- Organise visualisation of the results of those simulation, interpret them and select the most interesting to display in this thesis.

Chapter 2

State of the art

The behaviour of swarms and networks in collective decision-making has been studied extensively. Those studies are generally done on the best-of-n problem where agents must choose the best solution among a finite set of alternatives [Valentini et al., 2017] and use methods from statistical physics to analyse the system dynamics [Castellano et al., 2009]. In this work, we took a computational approach (using agent based simulations) because mathematical analysis is limited to (relatively) simple behaviours and using real robots would have taken too much time. We tested best-of-n problems with $n \geq 2$ and asymmetric and symmetric opinion quality with symmetric option cost (equal to 0). Moreover, we also tested the special case of symmetry breaking. [Castellano et al., 2009] review the literature concerning the use of statistical physics to study the dynamics of collective decision making in social systems. It details several points that will be used here such as the role of topology (see section 4.3), opinion dynamics and symmetry breaking.

2.1 Voter model rule

We use two models to process the neighbours information (see section 3.3). The first is the voter model, where the agent takes the opinion of one of its (randomly selected) neighbours. The classical voter model was theorised by Clifford and Sudbury in 1973 [Clifford and Sudbury, 1973] and named "voter model" in 1975 [Holley and Liggett, 1975]. Since then, it has been widely studied [Redner, 2019]. It was also tested on heterogeneous graphs [Sood and Redner, 2005] and networks [Sood et al., 2008]. An extension called "Weighted Voter Model" was proposed in [Valentini et al., 2014], it takes into account the agent's opinion quality (how good the agent thinks its answer is) and it will be used in this work (and be simply called voter model rule or VMR). Adding zealots to the classical voter model was first done in [Mobilia, 2003] concluding that in dimension $D \leq 2$ zealots could influence the results. The impact of having the same number of zealots on each side has been studied in [Mobilia and Georgiev, 2005] and [Mobilia et al., 2007]. In [Xie et al., 2011] the authors have demonstrated the presence of a tipping point of a percentage of zealots in the population (approximately 10% in a fully connected graph) at which the swarm quickly converges to a consensus in favour of the zealots opinion. This was tested on complete, Erdos-Rényi and Barabasi-Albert graphs using direct switch to update agents beliefs.

2.2 Local majority rule

The second model to process the neighbours information is the local majority rule (LMR) that was proposed in [Galam, S., 2002] to describe public debates. In this model, an agent pools its neighbours (agents it can communicate with) and take the majority's opinion. Its dynamics were studied in [Krapivsky and Redner, 2003] on a Two-State Interacting Spin Systems and using direct switch as update model. This model has known several extensions. The first one used moving agents [Galam et al., 2002] [Stauffer, 2002]. Other extensions include [Tessone et al., 2004] where each agent interacts with a variable number of neighbours and [Gekle et al., 2005] where it is extended to the best-of-3 problem. The impact of hostile agents such as contrarian (agent voting for the minority opinion) and zealots on the local majority rule model has also been studied. In [Galam, 2004] the author showed that when the proportion of contrarian is low there is a stable mixed phase with a clear majority–minority splitting. When the number of contrarians pass a threshold, there is a phase transition into a new disordered phase where no opinion dominates (the agents cannot break the symmetry). In [Stauffer and Martins, 2003] the authors have reproduced these dynamics for agents diffusing on a lattice and found that the phase transition only occurs at a higher agent density. In [Galam and Jacobs, 2007] the authors showed that adding zealots shift the separator (initially at 50%) towards their side and that having the same number of zealots on each side cancels out. Moreover, having more than 17% of zealots makes a side sure to win (if the opposing side does not have any zealots) while having more than 25% (equal number) of zealots in each side prevents the breaking of the symmetry. Finally, in [De Masi et al., 2021] the authors have show that when using VMR or LMR having a small proportion of informed agents (agents that can estimate the quality of its opinions) help to reduce the impact of zealots.

2.3 Models to update the agents opinion and strategies

The VMR and LMR models were associated with two models to update the agent opinion to form strategies (see section 3.3 to 3.5). Some of those strategies have already been compared between each other in the field of swarm intelligence. In the first model, an agent simply update its opinion to the new option (direct switch (DS) see section 3.2). In the second model, when an agent change opinion it goes to an uncommitted state, it is called cross-inhibition, when in an uncommitted state it takes the next opinion it is presented with (see section 3.2). Cross-inhibition (CI) was first observed in bees in [Seeley et al., 2012] [Pais et al., 2013] and later employed in swarm robotics in [Reina et al., 2015b] [Reina et al., 2015a]. In [Reina et al., 2017] the authors used ordinary differential equations to show that cross-inhibition can break the symmetry while direct switch does not. Cross-inhibition can also be used for value-sensitive response: break (quickly) the symmetry only if the quality value is higher than a threshold [Pirrone et al., 2022]. The impact of zealots on cross-inhibition and direct switch has been studied in [Prasetyo et al., 2020]. In [Canciani et al., 2019] the authors showed that cross-inhibition coupled with the voter model rule had better performances and was more resilient to contrarians than direct switch based strategies. We must first start by speaking of the strategies nomenclature. Indeed, in [Valentini et al., 2016a] the author tested the strategies we will call DS/LMR and DS/VMR. In their paper, those strategies are respectively called DMMD [Valentini et al., 2015] [Valentini et al., 2016b] and DMVD [Valentini et al., 2014]. DS/VMR was previously called weighted voter model but as cross-inhibition had not been used in collective decision-making yet the direct switch part was implicit. In this work it will thus be called DS/VMR for direct switch (weighted) voter model while DS/LMR will stand for direct switch local majority rule. Going back to [Valentini et al., 2016a], the author showed that DS/VMR is more accurate but slower than DS/LMR. They also tested a third strategy called DC (direct comparison of option quality) that has no modulation mechanism and allows the agents to share more information (quality estimates). They showed that the DC strategy is better for simple problem but does not scale to difficult problems. In [Biancalani et al., 2014] the authors focused on the behaviour of DS/VMR on symmetry breaking problem and showed that DS/VMR can only break the symmetry on easy problems (but the consensus is generally not stable). In [Talamali et al., 2021] it has been shown in that swarms where robots have fewer communication links (smaller communication radius or lower robots density) adapts better to new information even if this information is only observed by a minority of robots. [Talamali et al., 2019] showed that by varying how much the agents communicate over time (inspired by honeybee house-hunting) it is possible to find a better exploration versus exploitation balance and thus improve performances. [Trianni et al., 2016] and [Lerman and Galstyan, 2002] have shown that agents bumping into each other form cluster and that those cluster have an impact on swarms behaviour as it become heterogeneous. Moreover, collision can prevent population mixing. Thus, swarms with a lot of agents are experiencing less shuffling in reality than in a simulation that do not take this phenomenon into account. [Reina et al., 2018] have also showed this phenomenon make accurate simulation/modelling of swarm harder. The assumption of a well-mixed swarm is a limit to the simulation's accuracy.

2.4 Polarization

The creation of echo chambers is a common phenomenon in symmetry breaking. In such echo chambers, most agents share the same beliefs leading to a polarization of the swarm. This can slow down or prevent the breaking of the symmetry. This phenomenon was reproduced in [Starnini et al., 2016] by creating a model where agents with different opinions moved away and the opinion of close agents converged. [de Arruda et al., 2021] studied a model where agents were changed their neighbours favouring those with their opinion. They also studied the impact on contrarians and found that they would help the formation of echo chambers. In [Zhang et al., 2014] the authors studied the creation of echo chambers (opinion domains) on Random Geometric graph and the impact of zealots (committed agents). For their experiment, the authors put the same number of zealots in favour of each opinion. They showed that zealots have a huge impact on consensus time and tend to create opinion domains around them. They also showed that finite RG graph will be connected (does not contain isolated subgraphs) if the average degree is bigger than $ln(N)$. Moreover, echo chambers formation is a hot topic in social science studies for its impact on social media [Cinus et al., 2021] [Cinelli et al., 2021] [Baumann et al., 2020].

Chapter 3

Model

3.1 Structure of the simulator

The simulator allows the user to run experiments on a best-of-n problem with a number N of agents to study the process of collective decision-making. The goal is to reach a quorum of agents that have the same opinion on which is the best option among n alternatives (i.e. the option with the highest quality). Options' qualities have a value between 0 and 1 (see figure 3.2). Every agent stores two things: the option it deemed to be the best and an estimation of its quality, this estimation is used to set its probability to broadcast its opinion. At each step of a simulation (see figure 3.1), each agent receives social information in the form of messages from the agents it can communicate with, those agents are called neighbours. Moreover, the agents also gathers information from the environment, it has a probability to get a random option. If only one of the pieces of information (social or environmental) is available, the agent uses it to update its opinion. However, if both of them are available, it randomly chooses one and discards the other. If its opinion changed, the agents make a noisy estimate of the quality its new opinion before going into a latent state. While it is in the latent state, an agent does not receive or send any messages, nor interact with the environment. At each step, it has a probability to return to its normal (interactive) state. If it is not in latent state, it sends a message to its neighbours with a probability depending on the quality estimate of its opinion ($0.2 * \frac{\text{quality estimate}}{\text{max quality}}$), where 0.2 indicates the maximum communication frequency of an agent, i.e. 0.2 Hz.. Our work focuses on the impact different strategies to update the agent's opinion can have on the collective decision makings process and how this process is impacted when hostile agents are added.

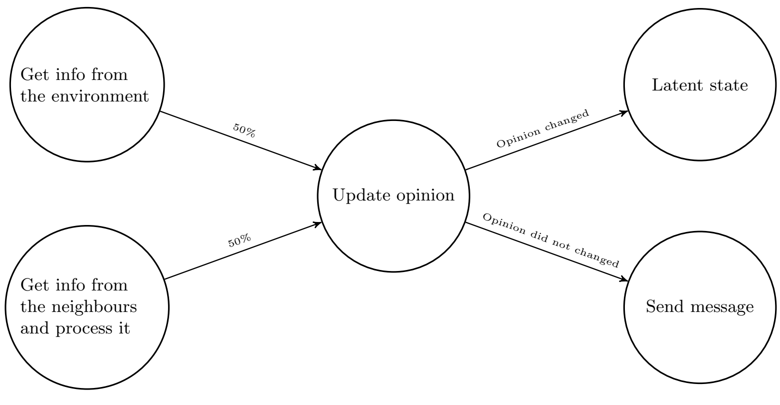


Figure 3.1: The operation of an agent during one step in the simulator.

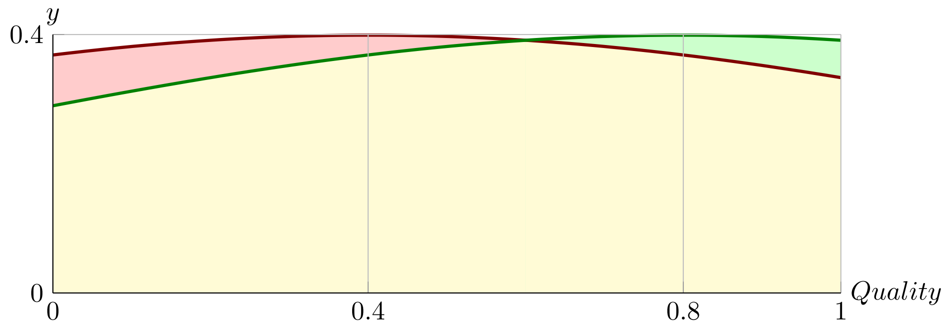


Figure 3.2: The probability density functions of two option having a value of 0.4 and 0.8 with a standard deviation value of 1 (it is trimmed as the minimum value is 0 and the maximum is 1).

3.2 Update the agent opinion

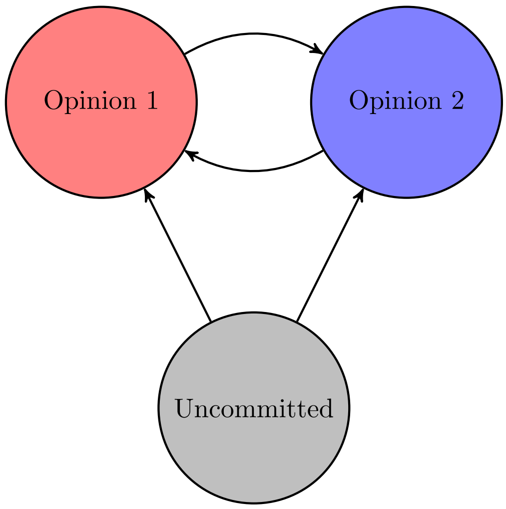


Figure 3.3: Direct switch

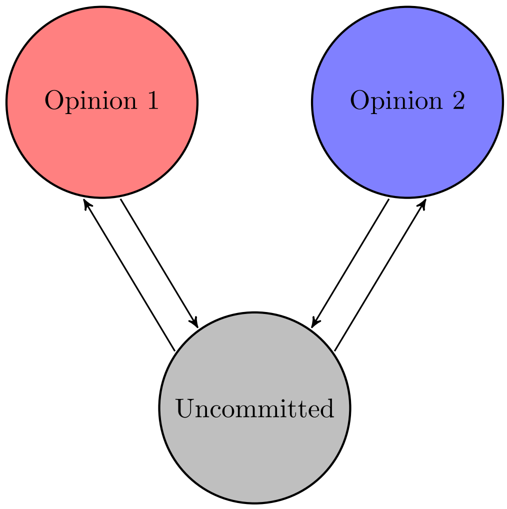


Figure 3.4: Cross-inhibition

We tested two rules to update the agent opinion (central state of figure 3.1). The first one is direct switch where an agent starts in the uncommitted state, and switches to the first opinion it finds (it considers this opinion as the best). If it is later given another opinion (by social or environmental input), the agent directly switches to it (figure 3.3), hence the name direct switch. The second rule is cross-inhibition, an agent following it starts uncommitted and switch to the first opinion it finds. However, if it is later given opinion (by social or environmental input), the agent return to the uncommitted state (figure 3.4). The process repeats until the swarm has reached a consensus.

3.3 Process the neighbours information

We tested two rules to allows an agent to process the information shared by its neighbours (other agents it can communicate with). The opinion resulting of this rule will be immediately inputted in the update opinion rule to update the opinion (see figure 3.1). The first rule to process the information shared by the neighbours is the voter model rule (VMR) where the agent chooses randomly one of its neighbours and picks its opinion. The second rule is the local majority rule (LMR) where the agent pools all its neighbours and picks the most frequently voted opinion.

3.4 Tested strategies

We tested four strategies, obtained by combining the two rules to update the agent opinion (direct switch and cross-inhibition) and the two rules to process the neighbours information (VMR and LMR). We did most of our tests using the best-of-2 problems. We compared the four strategies using three problem difficulties: an easy problem where the worst opinion's quality is half the best opinion's quality, a hard problem where the worst opinion's quality is 90% of the best opinion's quality, and a symmetry breaking problem where both opinions are of equal quality. The impact of several parameters on the decision-making process was tested. We obtained the results presented in this works by varying three parameters: the noise added to the agents estimation of the quality of their opinion, the agents communication radius (how many neighbours an agents has) and the number of possible option. When varying the last parameters the problem became a best-of-n problem with always one opinion significantly better than the other that were all similar (e.g., opinion A's quality = 0.4, opinion B's quality = 0.4, opinion C's quality = 0.8). Moreover, we also tested the four strategies on problems having a varying number of agents and on best-of-2 problems of increasing difficulty, but the previous parameters were judged insightful enough, so those results are not included in this work.

3.5 Hostile agents

In this work we studied the impact of hostile non-cooperative agents, those agent's goals can be to prevent, delay or sway the consensus. We tested three types of hostile agents. The first type of hostile agent is the zealot. A zealot never changes its own opinion and always shares it with all its neighbours, when we added zealots to a swarm, we made them all follow the worst opinion to see if they could influence the swarm into committing to a suboptimal opinion. The second type of hostile agent is the contrarian. A contrarian picks the less frequent opinion in its neighbourhoods. If a contrarian has no neighbour it behaves like a normal agent and can interact with the environmental information. The last type of hostile agent is the mad. At each time step, a mad agent picks randomly an opinion and gives it a random quality. When adding mad agents and contrarians, we expected that they would prevent, or at least slow down, consensus.

Chapter 4

Materials and methods

4.1 DeMaMAS

For this work we used and extended a simulator called DeMaMAS. The DEcision MAking Multi Agent Simulator (DeMaMAS) is a multiagent simulator for comparison of decentralised decision-making strategies written in python. It was developed at the University of Sheffield for studies of collective decision-making [University of Sheffield, 2020]. It was mainly developed to study how agents make consensus decisions when they have access to information from the environment in localized areas (called *totem*) and by communicating with other agents in their proximity. Indeed, at each step each agent calculates the distance to every other agents, if this distance is smaller than a given parameter, called the agent radius, the two agents can communicate with each other. For this master thesis, we improved the performance of DeMaMAS and expanded the range of experiments that it can conduct. Moreover, we decided to focus on the strategies the agents used to update their opinions and on the impact of hostile agents. Thus, in order to reduce the number of variables and obtain more general results, we decided that agents are able to access environmental information from everywhere on the torus, rather than being accessible only in localized are (now the *totems* cover the whole torus).

4.2 Optimisation of DeMaMAS

Two improvements were done on DeMaMAS, the first and simpler one was to limit the number of objects deep-copied. When the messages are sent simultaneously at the end of each time steps. In order to implement synchronous communication and avoid that the ordering with which agents' state is updated influences the dynamics, the original version of DeMaMAS deep-copied all the agent class at each iteration. This mechanism was not computationally efficient, now only the agent's message is copied. This optimization led to a significant speed-up.

The second optimisation was to speed up the computation of the list of neighbours. Indeed, each agent must know the list of its neighbours to communicate with them. This list depends on the communication range of the agents, a parameters called agent radius (or communication radius). The original code simply looped on all agents and computed the distance between all possible pairs of agents. The new code tests if the communication radius is smaller than a fourth of the environment side's length. If the communication radius is not small enough, no optimisation is possible, and the old algorithm is used. If the communication radius is small enough, the environment is divided into an x-by-x grid with squared cells. The grid's size (number of cells on one side) is $x = \lfloor \frac{environmentSize}{communicationRadius} \rfloor$. The agents are put in a list according to their position on the grid. Because each grid's cell is at least as large as the agent radius, the agent's neighbours can only be in the same cells or in one of the eight adjacent cells. Moreover, the fact that the distance used is symmetric implies that an agent is the neighbour of its neighbours. This fact allowed us to speed up the computation, as if in the figure 4.1 every agent of the cell 1 compute its distance with every agent of the cell 7, the agents of the cell 7 do not need to look in the cell 1 (the distance computation has already been done). Thus, it is possible to only compute the distance with the agents in the agent's cells, and in half of the adjacent cells. As a convention, we decided to use the cell above and the three cells on the right (the agents on the other cells will be the ones finding the current agent). However, the agent can go (and communicate) from one side of the environment to the other (like in a torus). Thus, the first and last column and the first and last row must be pasted in the grid at their opposite side. An example is provided in the figure 4.1.

	1	2	3	4	5	
25	21	22	23	24	25	21
20	16	17	18	19	20	16
15	11	12	13	14	15	11
10	6	7	8	9	10	6
5	1	2	3	4	5	1
	21	22	23	24	25	

Figure 4.1: The reproduction of the grid generated by a environment with $\lfloor \frac{environmentSize}{agentRadius} \rfloor = 5$. An agent in the red cell (number 7) has to compute the distance with the agents in its cell and in the blue cells. This process must be repeated for all the agents.

4.3 Expansion of DeMaMAS

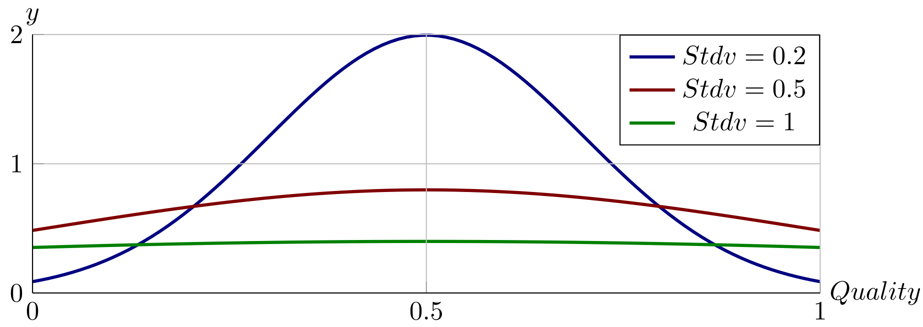
We added two features to DeMaMAS. The first one was to add a parameter that allowed to stop the simulation not when the quorum has been reached, but when it has been reached and maintained for a defined number of steps. This is useful to test the stability of the system. For these studies, we set this parameter to 500 steps. The second feature was the ability to run tests on three types of graphs. Each node of a graph represents an agent while each edge represents a communication link between two agents (i.e. two agents linked by an edge can communicate with each other). When running a simulation on those three graphs, the agents do not move, therefore their list of neighbours is fixed and set before the first step. DeMaMAS standard model (swarm) is a random geometric graph where the nodes moves over time and the communication links are thus recomputed at each time steps. It is a classical topology in swarm dynamics and has been used in many papers [Valentini et al., 2014] [Talamali et al., 2019] [Talamali et al., 2021]. Since the birth of graph theory with Euler's Seven Bridges of Königsberg problems [Euler, 1741], numerous models have been created. Here we choose to add three classical models to DeMaMAS. The first model is the random geometric graph that places the agents on DeMaMAS's torus (using a uniform distribution) and connects them if their distance is smaller to a parameter (called agent radius). In this model, there is a very high correlation between an agent's neighbours and its neighbours' neighbours. This model describes a very simple physical communication system: each agent communicates with all the other agents near him. The second model is one of the two

Erdős–Rényi (ER) models (also called Erdős–Rényi–Gilbert random graph model [Fienberg, 2012] or binomial graph). This model builds graphs by connecting nodes randomly. Each edge has a probability p to be included, p being independent of every other edge. Thus, the probability of generating each graph that has n nodes and M edges is $p^M(1-p)^{\binom{n}{2}-M}$ [Gilbert, 1959]. The third model is the Barabási–Albert (BA) model that generates scale-free networks. Thus, the generated networks have most of their nodes with a few edges and a few nodes (called hubs) with a very high number of edges (power law). This model was created to reproduce networks such as genetic networks, the World Wide Web or social networks [Barabasi and Albert, 1999]. The ER and BA graphs are implemented using the python library NetworkX [Hagberg et al., 2008].

4.4 Important Parameters

DeMaMAS simulations can be configured through a number of parameters, for an exhaustive list see its README file in the GitHub repository [University of Sheffield, 2020] and for all the value used see Appendix A. Here is a description of the most important parameters and the value used in this works. Most of them only had one value during the experiment but some of them had one or more base values (value used when varying the other parameters) and a set of values only tested once using the other parameters base values:
General parameters:

- environmentSize: float representing the dimension of the environment (the environment is a square). For this work it was set to 1.
- noiseStandardDeviationValue (Stdv): float representing the noise that the agent adds to its evaluation of the quality of an opinion. For this work its base value was 1. Moreover 0, 0.2 and 0.5 were also tested.
- numberOfAgents: the number of agents in the environment. For this work its base value was 100. Moreover 150, 200, 250 and 500 were also tested.
- numberOfSimulation: integer representing the number of simulations executed per configuration file. For this work it was set to 100.
- numberOfOpinion: integer representing the number of different opinion that can be shared by agents. For this work its base value was 2. Moreover 3, 5 and 10 were also tested.
- numberOfSteps: the duration of each simulation in time steps. For this work it was set to 25000.
- quorum: float representing the proportion of the swarm agreeing on one opinion needed to consider that there is a consensus. For this work it was set to 0.8.
- numberOfStepOverQuorum: the number of step there should be a consensus in the swarm to consider that the consensus is stable and finish the simulation. For this work it was set to 500.



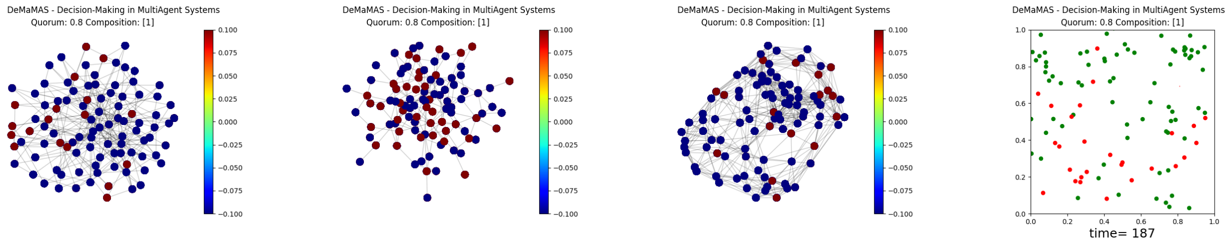
Behaviour parameters:

- maxQuality: it represents the maximum quality an opinion can have in the environment. For this work it was set to 1.
- qualityValues: vector of float representing the qualities assigned to the opinions. For this work base values were [0.4,0.8] (easy problem), [0.5,0.5] (symmetry breaking), [0.72,0.8] (hard problem). [0.6,0.8], [0.725,0.8], [0.75,0.8] and [0.775,0.8] were also tested. When using more than 2 opinions [0.4,...,0.4,0.8] was used. The easy and hard values were chosen to have the worst opinion's value be respectively 50 and 90 percent of the best opinion's value. [0.5,0.5] was simply chosen so the error is centred.
- updateModel: the rule used to update the agent opinion see section 3.2.
- updateRule: the rule used to process the neighbours information see section 3.3.

Agent characteristics parameters:

- numbeOfEdges: integer representing the number of edges to attach from a new node to existing ones (BA network). For this work its base value was 3. Moreover 4 and 5 were also tested.
- edgeProbability: float representing the probability of creating an edge between a new node and each existing ones (ER network). For this work its base value was 0.03. Moreover 0.04, 0.05, 0.1, 0.5 and 1 were also tested.
- agentRadius: float representing the agent communication-radius (RG network and swarm). For this work its base value was 0.2. Moreover 0.25, 0.3, 0.5 and 1 were also tested.

Base values for numbeOfEdges, edgeProbability, agentRadius were picked to be as low as possible while still creating connected graphs most of the time (if a graph is not connected, the code recreates it using a different random seed). The values were picked as low as possible to study the collective decision-making dynamics on a sparse network and appreciate differences between different types of networks. In fact, highly connected networks, in which nodes are connected to most of the other nodes, become similar to a fully-connected graph and make less evident the differences caused by the network topology. This means that the graphs have very different average degree (average number of nodes a node is connected to). BA has an average degree of 5.82, ER of $0.03 * (\text{ number of agents } - 1) = 2.97$, RG and swarm have an average degree of $0.2^2 * \pi * (\text{ number of agents } - 1) \approx 12.4$). Having RG this high was necessary to have a connected network [Zhang et al., 2014] while giving BA and RG such a high average degree may have resulted in behaviours very close to a fully connected networks. In retrospect, running experiments with all graph having an average degree equal to $ln(\text{ number of agents })$ (which is the minimum average degree to have connected RG graph) would have been interesting. An example of a graphs generated using every model used in the experiment with its base parameters can be seen in figure 4.2.



(a) Barabási-Albert (b) Erdős–Rényi (c) Random geometric (d) Swarm

Figure 4.2: (a) Barabási-Albert graph with a numbeOfEdges=3, (b) Erdős–Rényi graph with an edgeProbability=0.03, (c) Random geometric graph and (d) swarm with an agentRadius=0.2.

Movement parameters (only for swarm):

- moveDimension: float representing by how much an agent move at each step. For this work it was set to 0.005.
- straightLength: integer representing how many time step the agent needs to wait in order to change their direction. For this work it was set to 20.

Message parameters:

- msgType: represents the information shared by the agents, in this work ”simple” was used ie: the agents shared only their own opinion.
- sendConstant: represents how often the agents can send a message. For this work it was set to 1 ie: the agents could send a message at each time steps.
- sendInfoMethod: represent which method will be used by agents to share messages. For this work “wv” was used, this means that the agents shared their opinion with a probability related to its quality.

Interaction function parameters:

- preStepSelfStrength: probability of interacting with a randomly selected opinion after the condition in the interaction function is verified. For this work 0.2 was used.
- preStepSocialStrength: represents how much an agent can interacts with others after the condition in the interaction function is verified. For this work 1 was used (it can always interact).
- interactiveProb: each time an agent updates its opinion, it goes into a ‘latent’ state during which it does not receive nor send messages. This is the probability that a latent agent returns interactive (restart interacting with the environment/other agents) at each time step. This help reducing information cascades, correlation in neighbourhood between two consecutive steps, and in general improves the decision making performance [Reina et al., 2015b]. Moreover, it has the added benefit of slowing down the simulation it is easier to see differences between two runs. For this work 0.05 was used (an average of 20 time steps).

4.5 Graphical simulation

DeMaMas also supports graphical simulation to visualise the simulation step by step. It was extended to also be able to handle graphs.

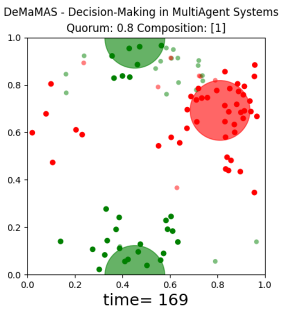


Figure 4.3: Original version of DeMaMAS were a swarm of agents interact with totems.

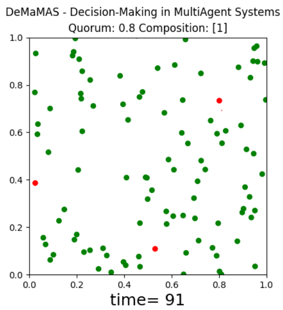


Figure 4.4: Current version of DeMaMAS were the agents can access information about the options from every position in the environments.

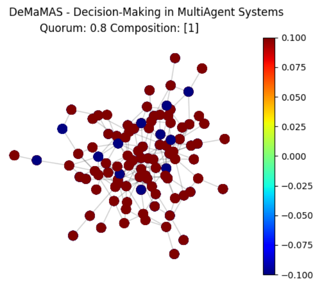


Figure 4.5: Current version of DeMaMAS were a graph (here ER) made of agents which can interact with the environment and communicate with other agents if they are linked by an edge.

4.6 Newmajorana Cluster

DeMaMAS was run on the High Performance Computer (HPC) of the IRIDIA Laboratory, comprising more than 1500 CPUs, and based on the SLURM job scheduler. Using batch scripts allowed the launching of hundreds of runs at the same time.

Chapter 5

Results

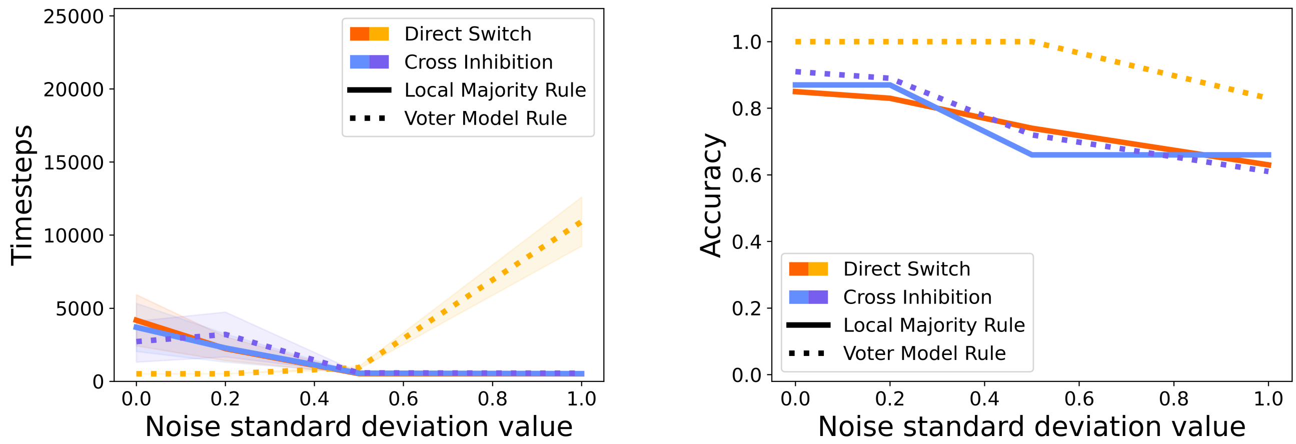
5.1 Comparison of the four methods in a fully cooperative environment

5.1.1 Easy problem with varying levels of noise

- DS/VMR has poor performances but does not commit to the wrong solution
- other strategies behave quite similarly
- random geometric network (RG) is slower and less accurate

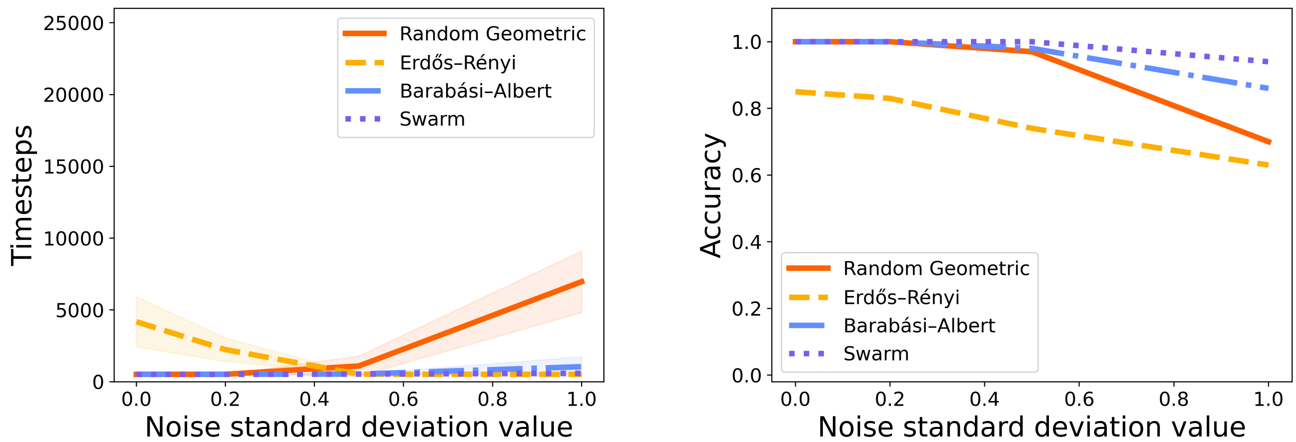
The voter model rule (VMR) is known to be slower but more accurate than the local majority rule [Valentini et al., 2016a]. Indeed, figures 5.2 to 5.5 show that direct switch with voter model rule (DS/VMR) is the slowest strategy. Moreover, since DS/VMR does not always commit to a solution in a reasonable time its accuracy is generally low. The decrease in accuracy for time-consuming problems is caused by excessively long convergence time, however, in the (tens of) thousands of runs analysed DS/VMR did not commit to the wrong solution once, this can be an interesting property for very specific applications. A surprising result is that it is the only strategy that performs worse on very connected networks, that is, when there is a high number of agents (except for ER) or when the network connectivity parameter determining the number of neighbours is high. In the literature, VMR and LMR have generally been tested alongside direct switch (as cross-inhibition is much younger). Using cross-inhibition with VMR (CI/VMR) gives better results than with DS/VMR. While results for CI/LMR are similar to those obtained using DS/LMR. To summarise, DS/VMR should be used exclusively on easy problems, or where accuracy is the most important factor regardless of the high convergence time. Instead, the other three strategies behave quite similarly, with none being the best in all situations.

When testing the different networks, random geometric network appear far slower than the other networks for every strategy that is using LMR. It also appear slower on CI/VMR (figure 5.5) while not being the worst on DS/VMR (figure 5.3). Its accuracy is not very good either. It is interesting to note that the swarm network, which is a random geometric network (RG) with slow mixing lead to a better collective behaviour except for DS/VMR where it lead to a worst collective behaviour. This can be explained by the fact that agents in RG have lots of neighbours in common with their neighbours creating regions where one belief dominate (echo chambers). This characteristic also makes information spreading slower as it cannot go from one side of the arena to the other (all neighbours of an agent are within a range). The dynamics on ER network are slower for easy problems with low standard deviation (figure 5.1). To summarise, apart from the fact that the dynamics on random geometric networks are slower, it is hard to see a clear trend. The slowness of random geometric networks can be due to the fact that information only spreads in a geometric way like thermal conduction while in the swarm network, moving robots can be interpreted as some sort of convection even if the robots move randomly and not according to their opinion. This analogy could be an explanation for why the convergence on swarm network is faster than on random geometric network.



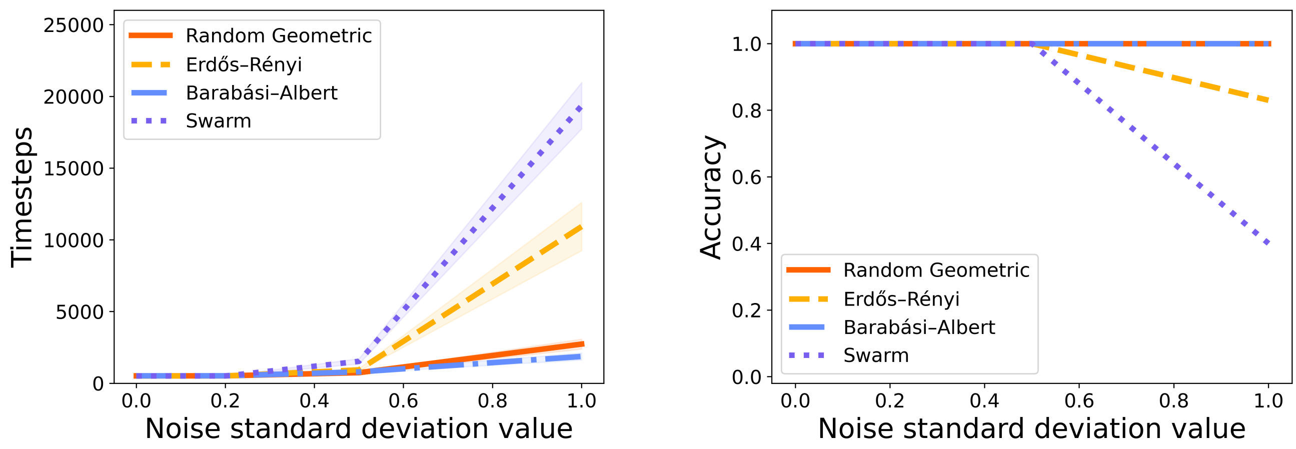
(a) Timesteps (b) Accuracy

Figure 5.1: Timesteps and accuracy for an easy problem on an Erdős–Rényi network with the standard deviation going from 0 to 1.



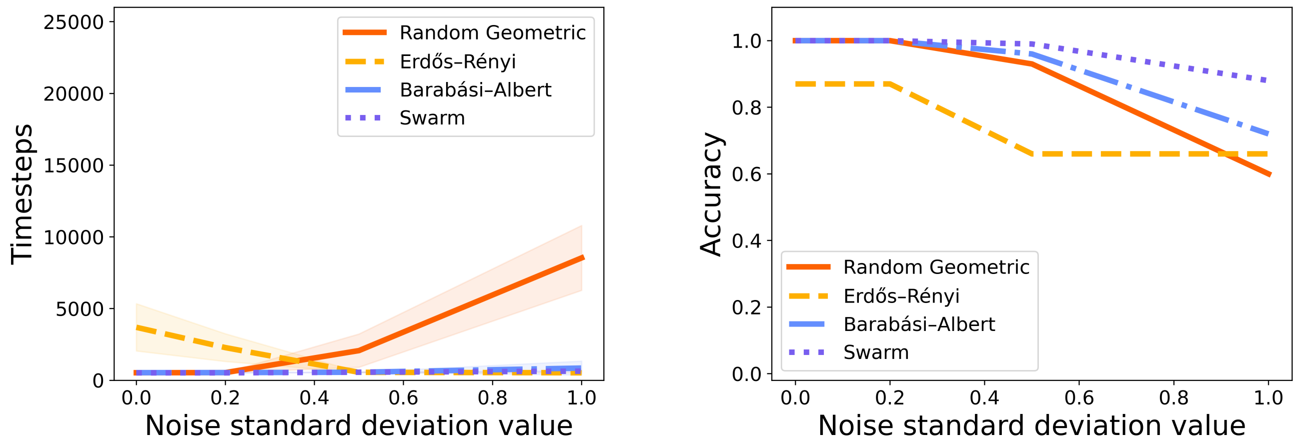
(a) Timesteps (b) Accuracy

Figure 5.2: Timesteps and accuracy for an easy problem using direct switch and local majority rule with standard deviation going from 0 to 1.



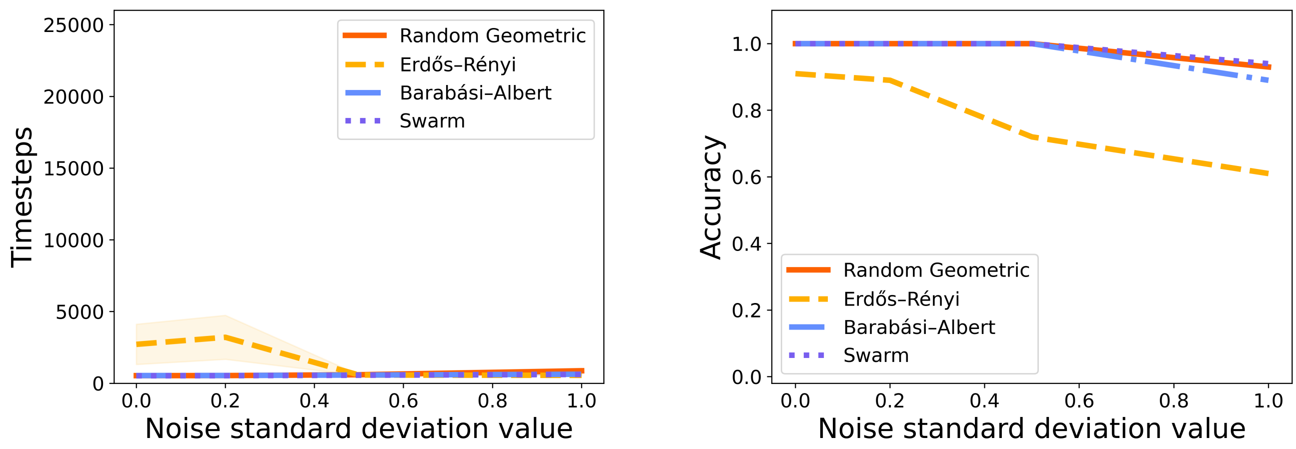
(a) Timesteps (b) Accuracy

Figure 5.3: Timesteps and accuracy for an easy problem using direct switch and voter model rule with the standard deviation going from 0 to 1.



(a) Timesteps (b) Accuracy

Figure 5.4: Timesteps and accuracy for an easy problem using cross-inhibition and local majority rule with the standard deviation going from 0 to 1.



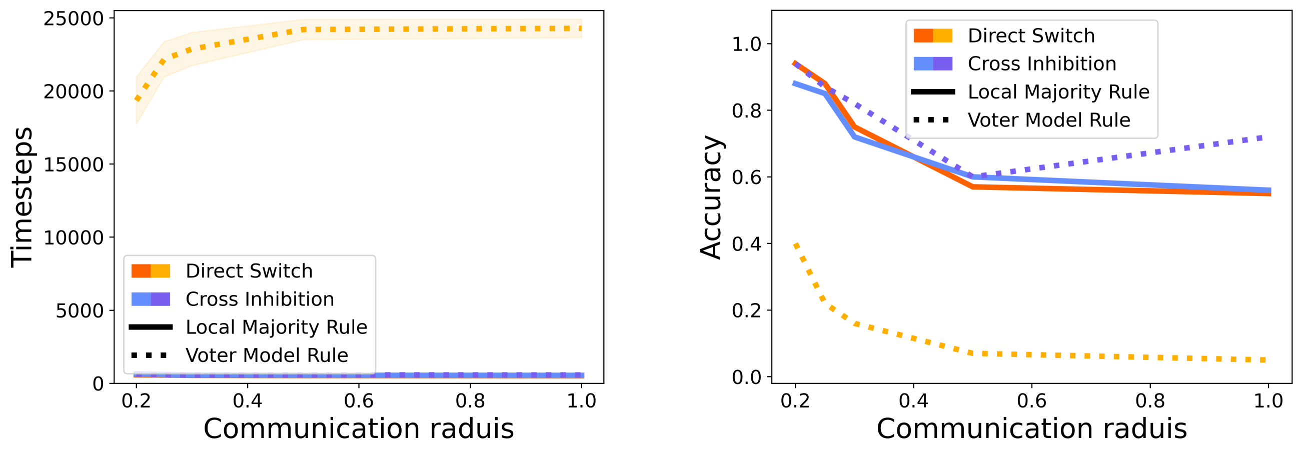
(a) Timesteps (b) Accuracy

Figure 5.5: Timesteps and accuracy for an easy problem using cross-inhibition and voter model rule with the standard deviation going from 0 to 1.

5.1.2 Easy problem with varying agent's communication radius

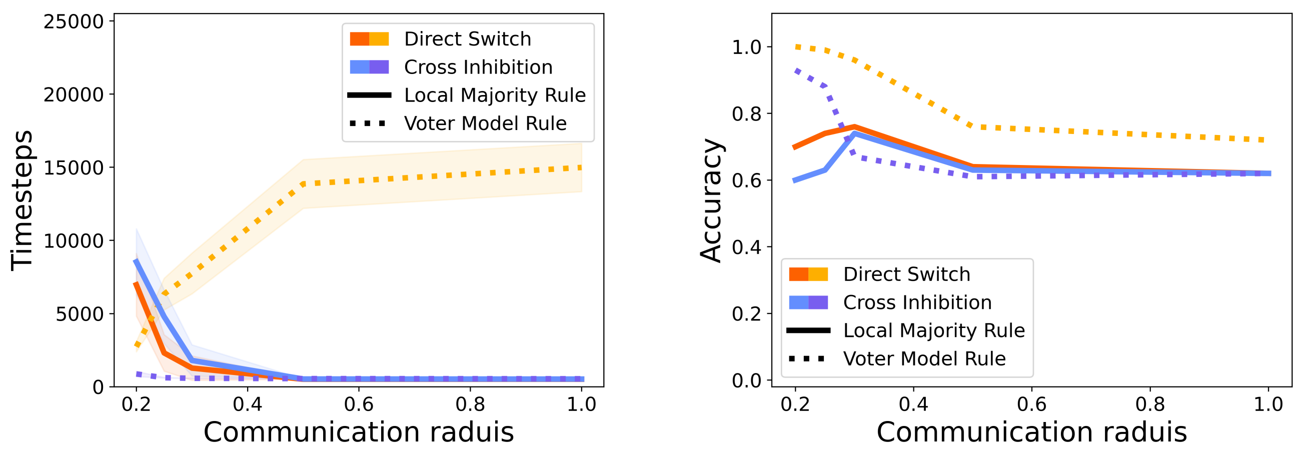
- increasing the average degree speeds up the process
- VMR-based strategies on networks with high average degree can lead to suboptimal choices

Figure 5.6 shows that increasing the communication raduis (and thus the average degree) makes the problem easier, however the results for the swarm network are not as clear as the other networks because the basic problem is too easy. However, for random geometric graph the time needed to reach a stable a consensus decreases as the agent radius increases (figure 5.7). This stops when the agent radius is equal to 0.5. At this point, the time taken is close to 500 steps (the minimum number of steps required to reach a stable consensus). However, for the swarm the accuracy decreases as the agent radius increases. This is also true for the VMR-based strategies on RG graphs. However, the accuracy of LMR-based strategies does not seem to be affected. Similar results are obtained for other networks.
We can thus conclude that increasing the average degree speeds up the process but can lead to suboptimal choices. When increasing the number of agents in all types of graphs the time needed to reach a stable consensus also decreases but the accuracy generally seems to slightly increase (except for DS/VMR). It seems that with the current parameters and a (nearly) fully connected networks the swarm commits too quickly to an opinion without really testing if it is really the best. Moreover, if a swarm uses the local majority rule (LMR) and reaches a large majority in favour of an opinion it is very difficult for a better opinion in a minority to prevail. This is consistent with the literature, indeed in [Talamali et al., 2021] the authors showed that LMR-based strategy are bad at reacting to changes (they cannot switch opinion if a new better opinion appears). They also showed that having fewer communication links allows a swarm to better adapt to changes.



(a) Timesteps (b) Accuracy

Figure 5.6: Timesteps and accuracy for an easy problem on a swarm network with the agent radius going from 0.2 to 1.



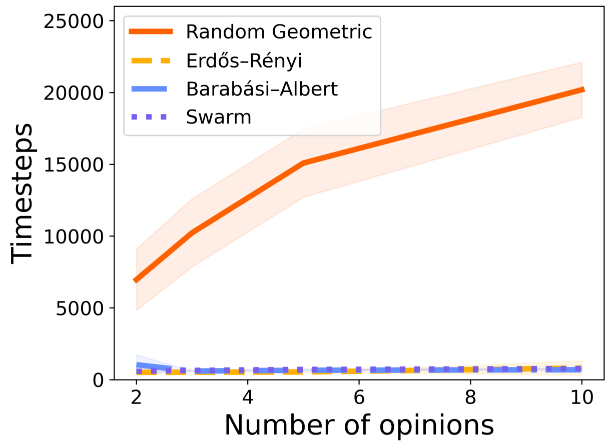
(a) Timesteps (b) Accuracy

Figure 5.7: Timesteps and accuracy for an easy problem on a random geometric graph with the agent radius going from 0.2 to 1.

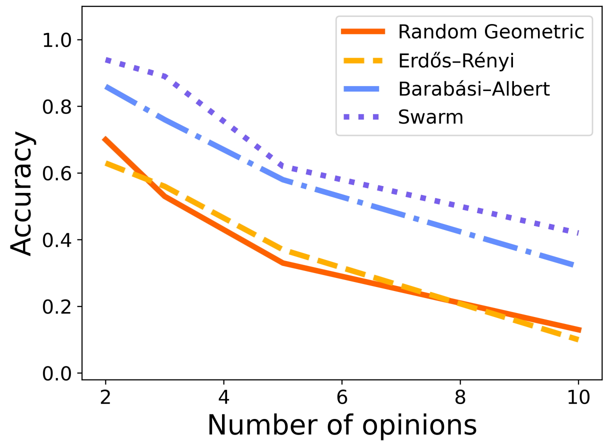
5.1.3 Easy problem with varying number of options

- adding options decreases accuracy and slow down the decision making process

For every combination of strategy/type of network, adding options decreased accuracy (figures 5.8 to 5.11). This can be explained by the fact that operating in environments with a large number of options reduces the probability to interact with the best option. Indeed, at each step, each agent has a 20% chance of interacting with the environment (and a 50% chance of using the environmental information if the agent also interacts with its neighbours). If it does, it will interact with a randomly chosen option (see section 3.1). Thus, adding options reduces the chances of interacting with the best one. figures 5.9 to 5.11 also shows that adding options also increase noticeably the time taken to converge to a consensus in random geometric graphs and figure 5.8 shows that if a swarm use DS/LMR, adding options will also increase noticeably the time taken to converge to a consensus. For random geometric this could be explained by the presence of clusters of agents with the same opinion (echo chambers), adding more options might increase the number of echo chambers and thus slow down the decision-making process. Moreover, figure 5.2 shows that increasing the problem's difficulty has a big impact on DS/LMR performances.

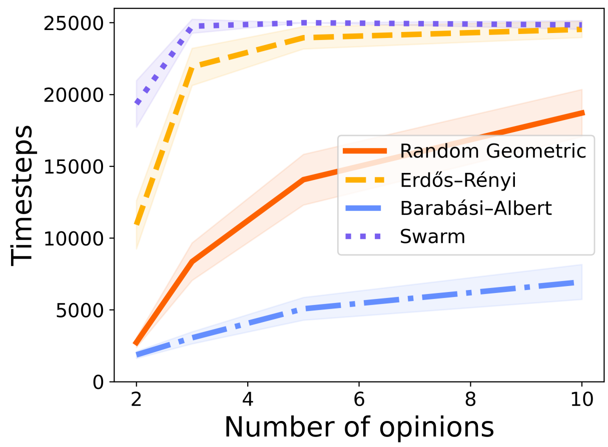


(a) Timesteps

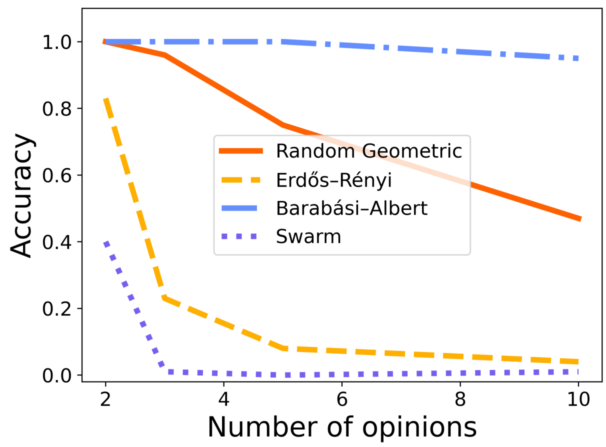


(b) Accuracy

Figure 5.8: Timesteps and accuracy for an easy problem using direct switch and local majority rule with the number of options going from 2 to 10.

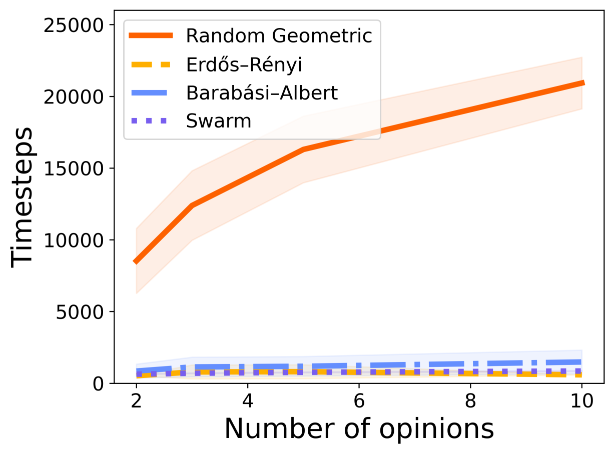


(a) Timesteps

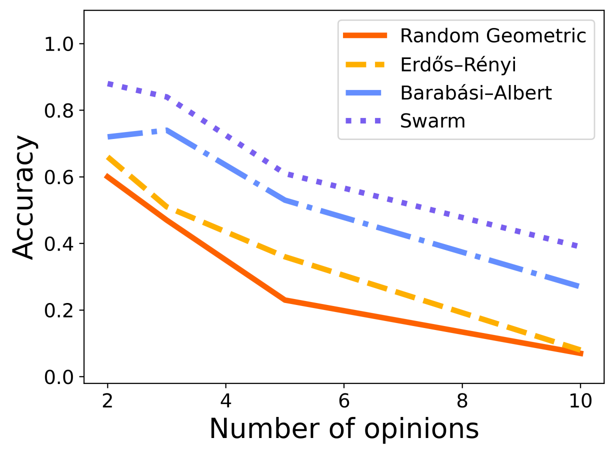


(b) Accuracy

Figure 5.9: Timesteps and accuracy for an easy problem using direct switch and local voter model rule with the number of options going from 2 to 10.

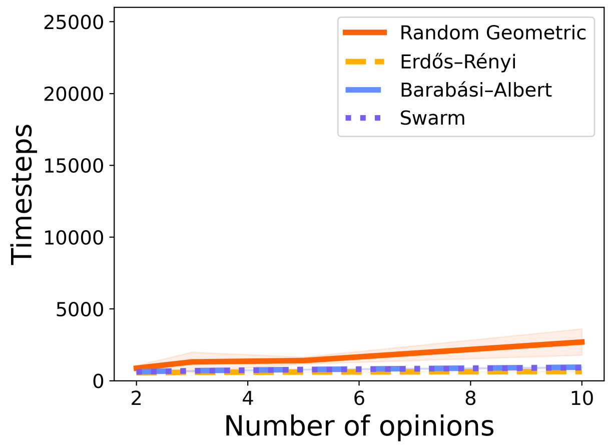


(a) Timesteps

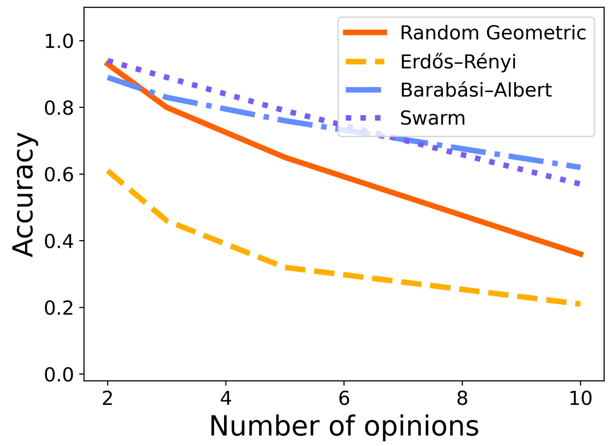


(b) Accuracy

Figure 5.10: Timesteps and accuracy for an easy problem using cross-inhibition and local majority rule with the number of options going from 2 to 10.



(a) Timesteps



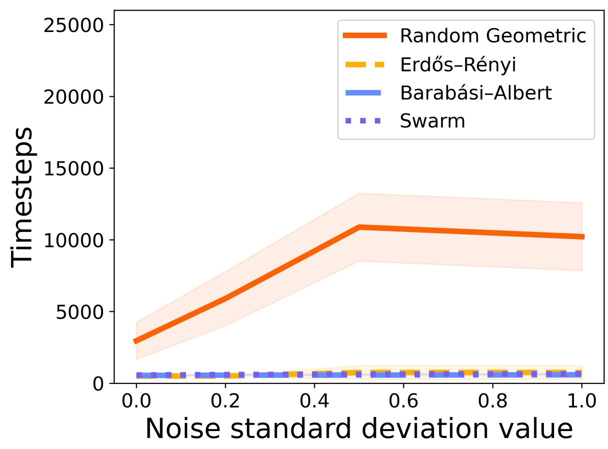
(b) Accuracy

Figure 5.11: Timesteps and accuracy for an easy problem using cross-inhibition and voter model rule with the number of options going from 2 to 10.

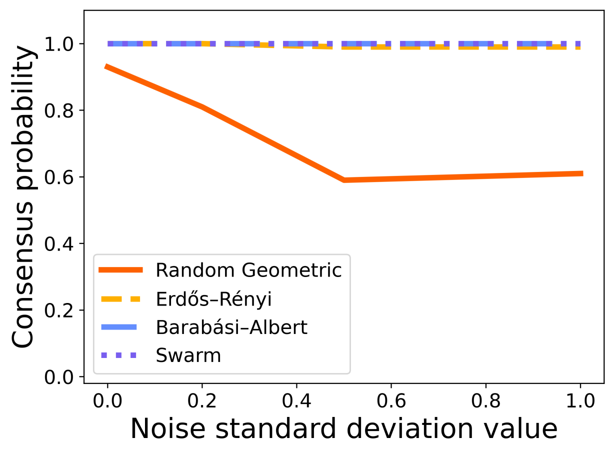
5.1.4 Symmetry breaking problem with varying levels of noise

- DS/VMR is not able to establish stable consensus in the symmetry breaking problem
- RG is the topology with the worst results in the symmetry breaking problem

The figures 5.12 to 5.15 show that only two parameters prevent achieving a consensus in symmetry breaking problems when the noise's standard deviation is going from 0 to 1. Those parameters are the use of the DS/VMR strategy and the use of the random geometric network (similar results are obtained when varying other parameters). The fact DS/VMR is not able to establish stable consensus in the symmetry breaking problem is consistent with the literature [Biancalani et al., 2014] [Reina et al., 2017]. While the poor results of RG can be explained by the very high correlation between an agent's neighbours and its neighbours' neighbours. This can create several geographical different echo chambers where one option make consensus, slowing down the overall decision-making process. Every other combination of strategy/type of network nearly always reached a stable consensus in less than 25000 steps.

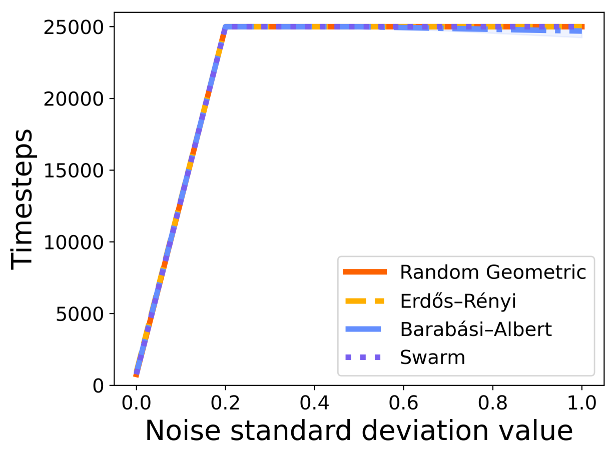


(a) Timesteps

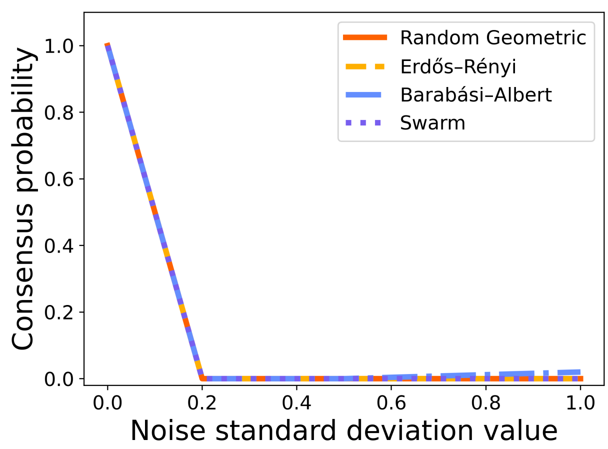


(b) Consensus probability

Figure 5.12: Timesteps and consensus probability for a symmetry breaking problem using direct switch and local majority rule with the standard deviation going from 0 to 1.

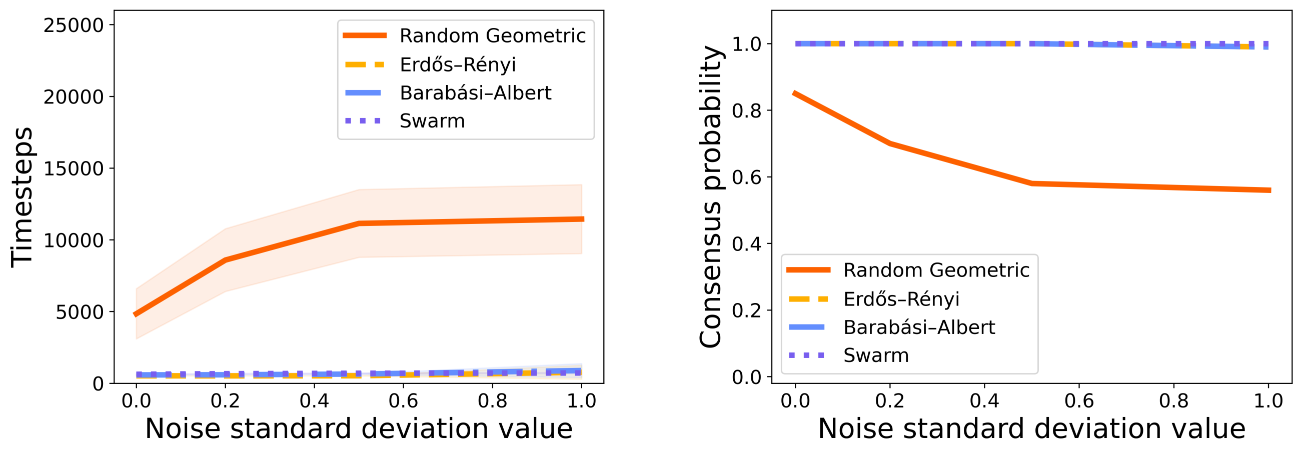


(a) Timesteps



(b) Consensus probability

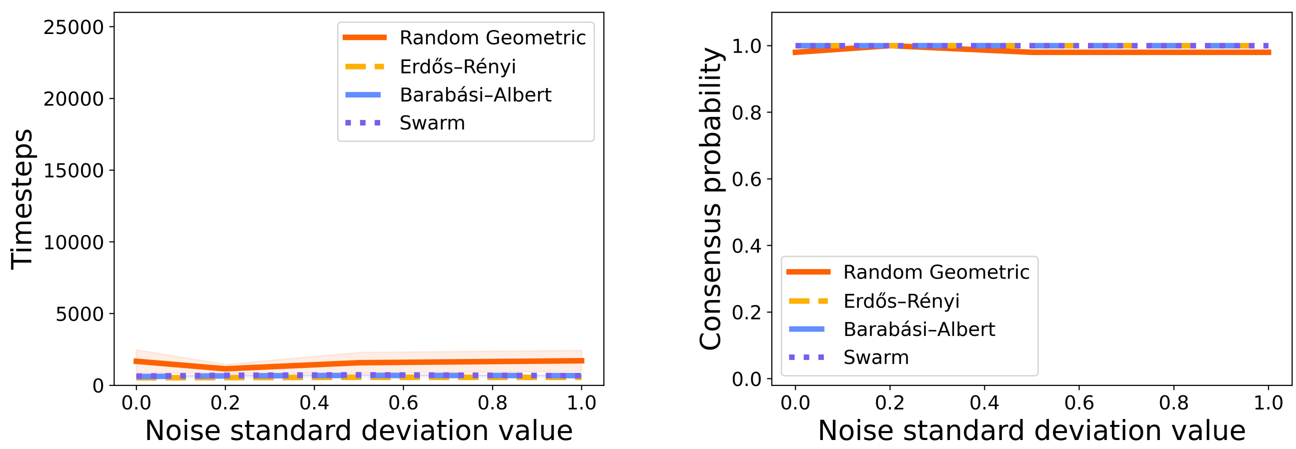
Figure 5.13: Timesteps and consensus probability for a symmetry breaking problem using direct switch and voter model rule with the standard deviation going from 0 to 1.



(a) Timesteps

(b) Consensus probability

Figure 5.14: Timesteps and consensus probability for a symmetry breaking problem using cross-inhibition and local majority rule with the standard deviation going from 0 to 1.



(a) Timesteps

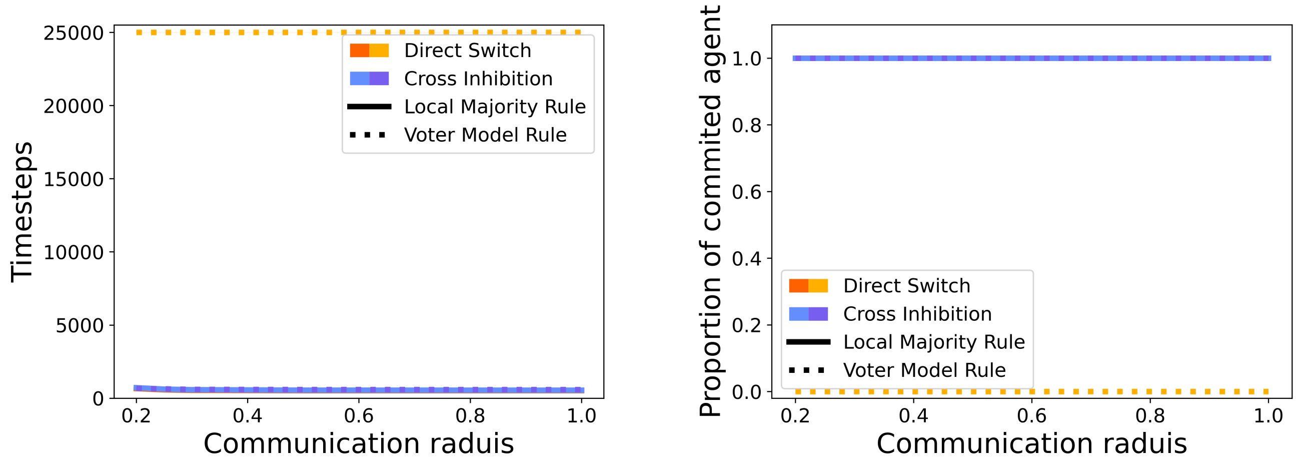
(b) Consensus probability

Figure 5.15: Timesteps and consensus probability for a symmetry breaking problem using cross-inhibition and voter model rule with the standard deviation going from 0 to 1.

5.1.5 Symmetry breaking problem with varying agent's communication radius

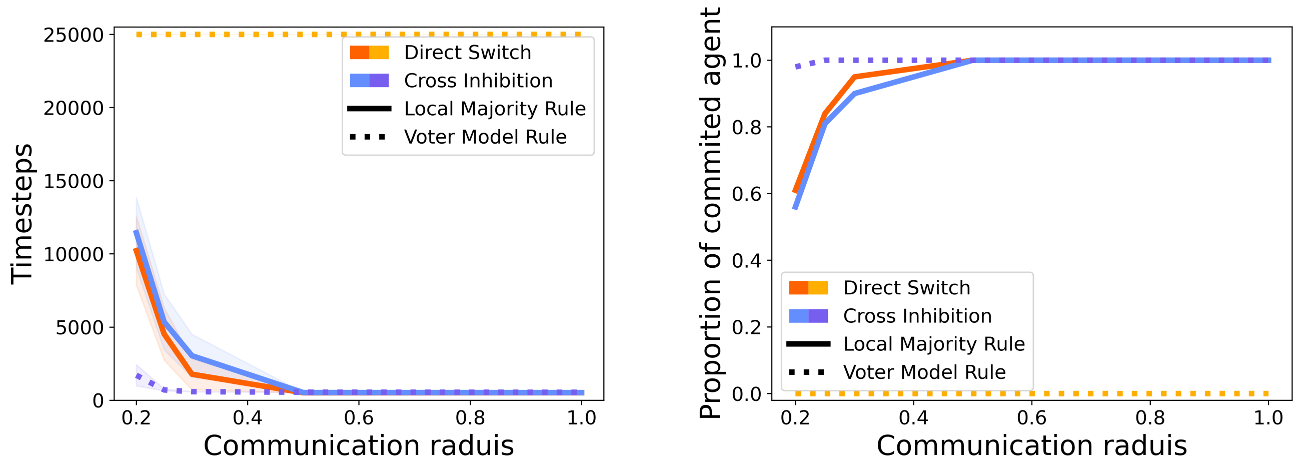
- increasing the communication radius speeds up the breaking of the symmetry
- CI/VMR has very good results

Like for the easy problems having increasing average degree helps to reach a stable consensus faster (figures 5.16 and 5.17). However, the consensus probability also increases. Thus, having access to more information makes the symmetry breaking problem easier. It is also interesting to note that LMR-based strategies have worse results than CI/VMR, this is a surprising fact and would be interesting to investigate.



(a) Timesteps (b) Consensus probability

Figure 5.16: Timesteps and consensus probability on a swarm with the agent radius going from 0.2 to 1.



(a) Timesteps (b) Consensus probability

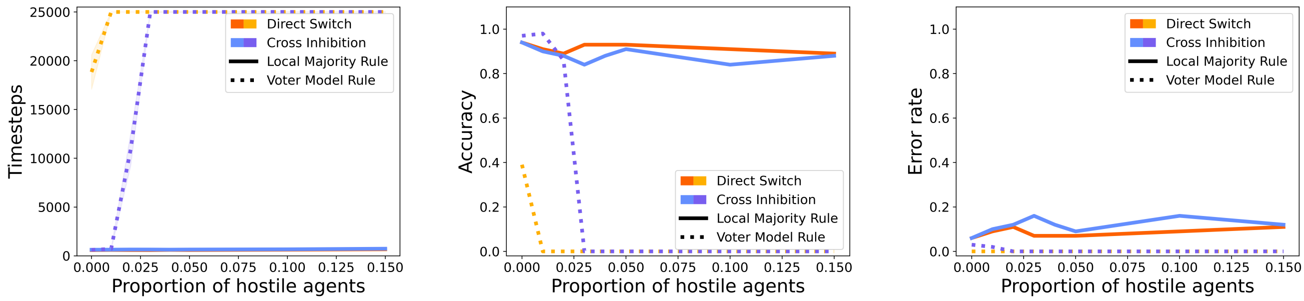
Figure 5.17: Timesteps and consensus probability on a random geometric graph with the agent radius going from 0.2 to 1.

5.2 Comparison of the four methods in a non-cooperative environment

5.2.1 Easy problem with varying number of hostile agents

- the VMR-based strategies are very sensible to mad agents, zealots
- LMR-based strategies are only sensible to zealots
- contrarians only effect VMR-based in certain conditions
- BA is the most resilient network topology

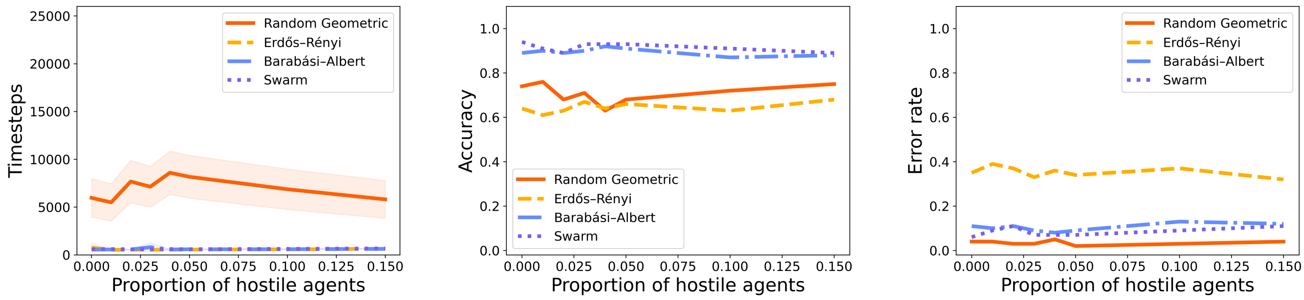
The mad agents impact strongly the VMR-based strategies by reducing the probability to reach a stable consensus (figure 5.18). However, it does not impact much the probability of choosing the wrong opinion. The fact that LMR-based strategy are less impacted could be explained by the fact that it takes the most popular opinion over a neighbourhood. Thus, mad voter statistically cancels each other out by voting for both options with the same probability.



(a) Timesteps (b) Accuracy (c) Error rate

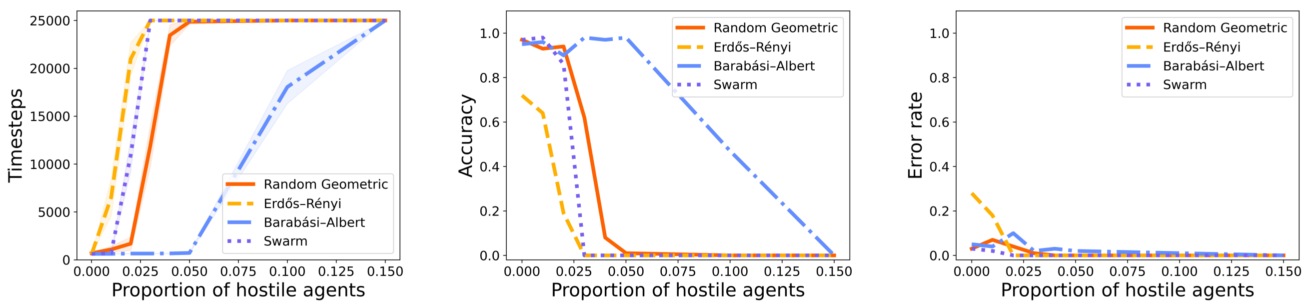
Figure 5.18: Timesteps, accuracy and error rate (proportion of agents committing to the wrong opinion) for an easy problem on a swarm with the proportion of mad agent going from 0 to 0.15.

BA is the most accurate graph while ER is the worst (figures 5.19 and 5.20). Swarm is faster than RG but has a similar accuracy. These results suggest that having hubs helps to resist mad agents. Moreover, the mixing provided by the swarm over the RG helps to accelerate the decision-making process.



(a) Timesteps (b) Accuracy (c) Error rate

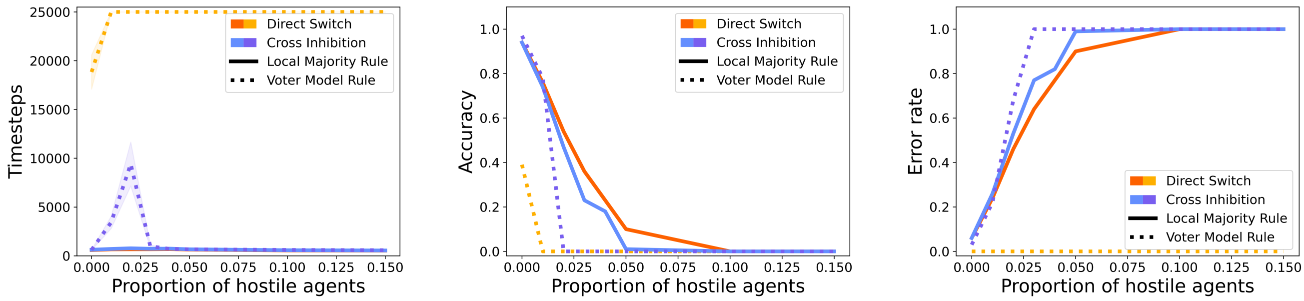
Figure 5.19: Timesteps, accuracy and error rate (proportion of agents committing to the wrong opinion) for an easy problem using direct switch and local majority rule with the proportion of mad agent going from 0 to 0.15.



(a) Timesteps (b) Accuracy (c) Error rate

Figure 5.20: Timesteps, accuracy and error rate (proportion of agents committing to the wrong opinion) for an easy problem using cross-inhibition and voter model rule with the proportion of mad agent going from 0 to 0.15.

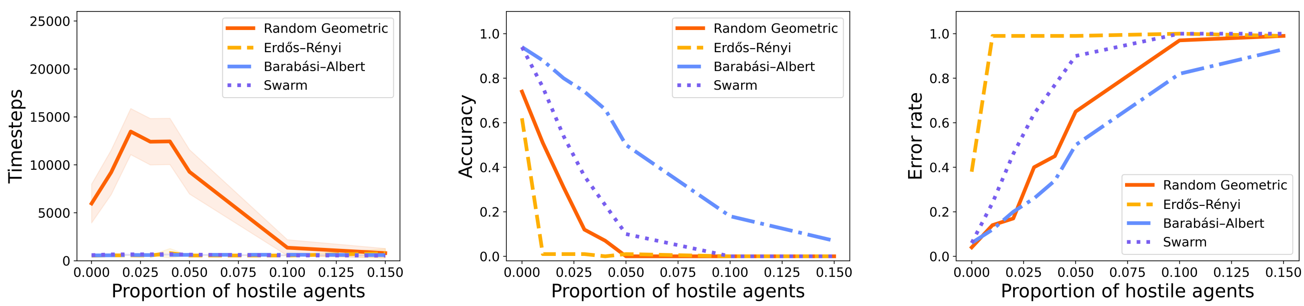
The figure 5.21 shows that zealots have a big impact on the accuracy and the error rate (except for DS/VMR that do not reach any consensus when zealots are introduced). They generally do not increase the Timesteps except for certain configurations where the consensus times have inverted V shape. In the first part, the consensus times rise and agent no longer reach a consensus. In the second part of the inverted V shape, the consensus times drop and the agents reach a consensus in favour of the inferior (but supported by the zealots) option.



(a) Timesteps (b) Accuracy (c) Error rate

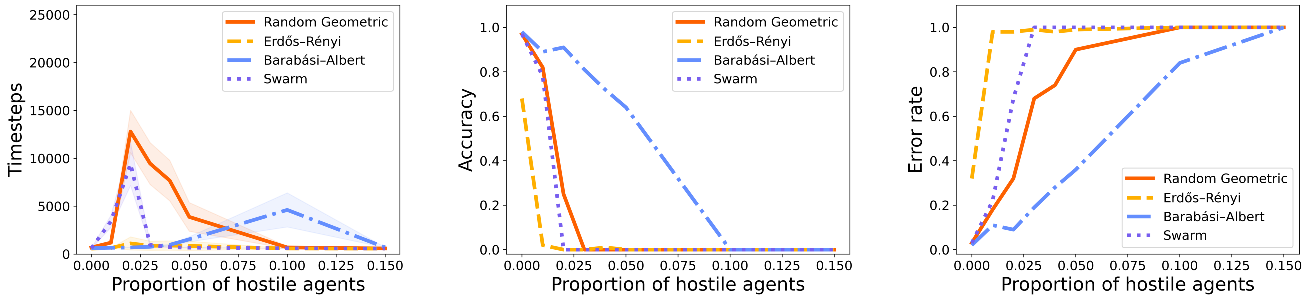
Figure 5.21: Timesteps, accuracy and error rate (proportion of agents committing to the wrong opinion) for an easy problem on a swarm with the proportion of zealots going from 0 to 0.15.

The results are similar to those obtained with mad agent: BA is the most accurate while ER is the less accurate (figures 5.22 and 5.23). Swarm is faster than RG but has a similar accuracy. It seems that having hubs also helps to resist zealots and that the mixing provided by the swarm over the RG helps to accelerate the decision-making process.



(a) Timesteps (b) Accuracy (c) Error rate

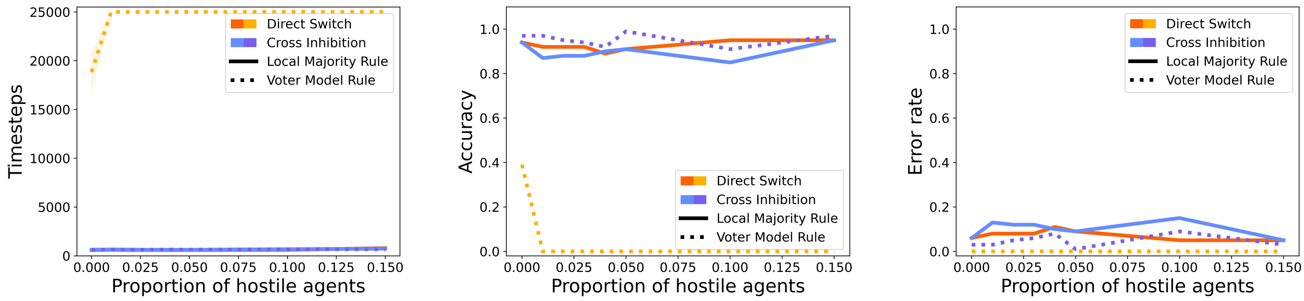
Figure 5.22: Timesteps, accuracy and error rate (proportion of agents committing to the wrong opinion) for an easy problem using direct switch and local majority rule with the proportion of zealots going from 0 to 0.15.



(a) Timesteps (b) Accuracy (c) Error rate

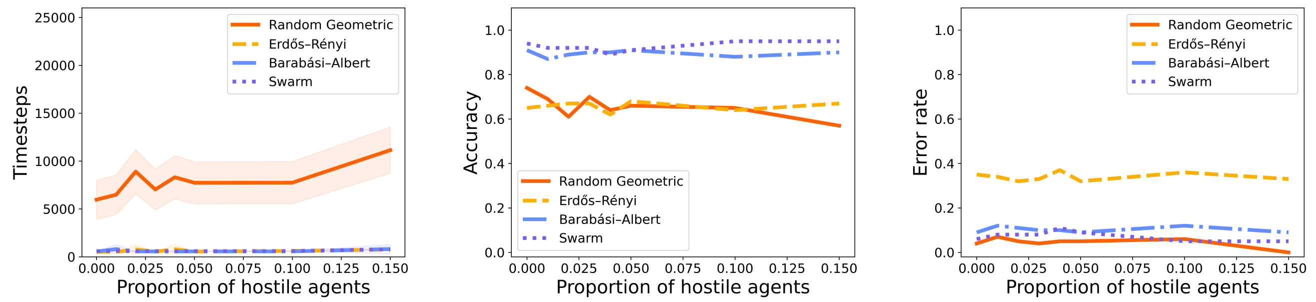
Figure 5.23: Timesteps, accuracy and error rate (proportion of agents committing to the wrong opinion) for an easy problem using cross-inhibition and voter model rule with the proportion of zealots going from 0 to 0.15.

Surprisingly the contrarians do not seem to have much effect, this results was expected for majority based strategy (figure 5.25). Indeed, if there is a clear majority, adding one or two agents to the minority will not change the final results: the majority's opinion will be chosen by every cooperative agent. However, the fact that CI/VMR is not strongly impacted is surprising (figure 5.26). In fact, the only (small) impact we could see is on ER networks and CI/VMR and for more than 5% of contrarians. It might be due to their low average degree that might cause certain agents to only have contrarian neighbours. Surprisingly this slow down seems to improve the accuracy. This phenomenon will also be encountered in subsection 5.2.2. For majority-based strategies (figure 5.24) only RG is impacted it might be due to its higher sensibility to hard problems as it was shown by its behaviour in a fully cooperative environment with the noise varying (see subsection 5.1.1).



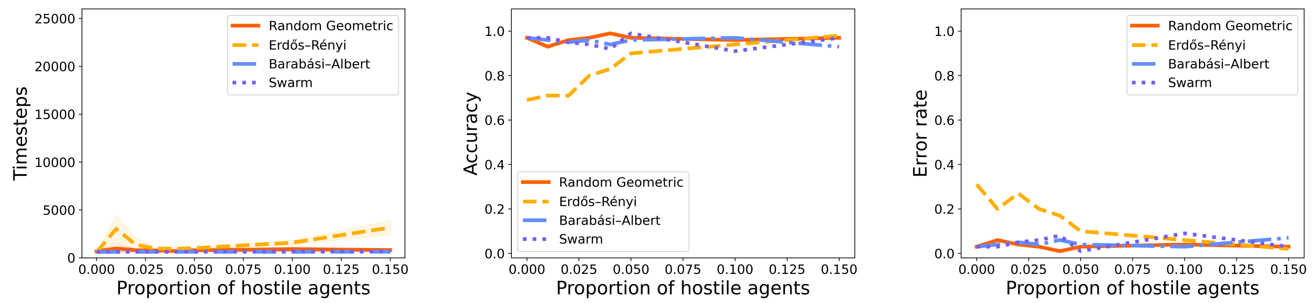
(a) Timesteps (b) Accuracy (c) Error rate

Figure 5.24: Timesteps, accuracy and error rate (proportion of agents committing to the wrong opinion) for an easy problem on a swarm with the proportion of contrarians going from 0 to 0.15.



(a) Timesteps (b) Accuracy (c) Error rate

Figure 5.25: Timesteps, accuracy and error rate (proportion of agents committing to the wrong opinion) for an easy problem using direct switch and local majority rule with the proportion of contrarians going from 0 to 0.15.



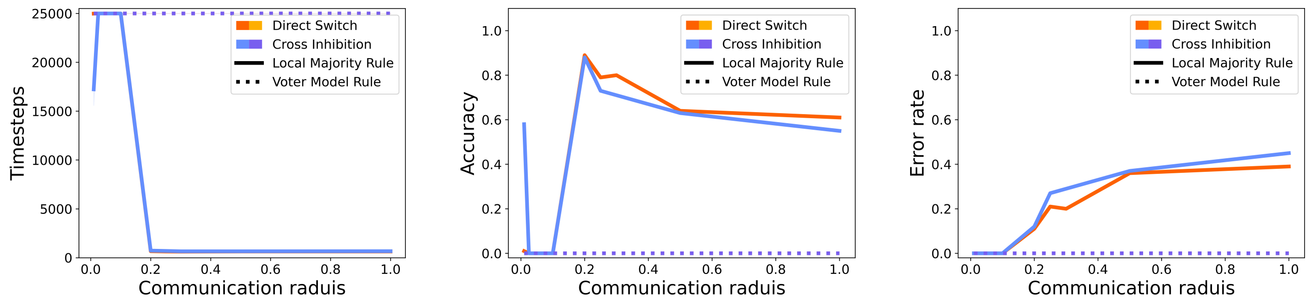
(a) Timesteps (b) Accuracy (c) Error rate

Figure 5.26: Timesteps, accuracy and error rate (proportion of agents committing to the wrong opinion) for an easy problem using cross-inhibition and voter model rule with the proportion of contrarians going from 0 to 0.15.

5.2.2 Easy problem on a swarm with varying agent's communication radius

- high enough communication radius allows VMR-based strategies to ignore mad agents and contrarians
- contrarians can improve CI/VMR performances

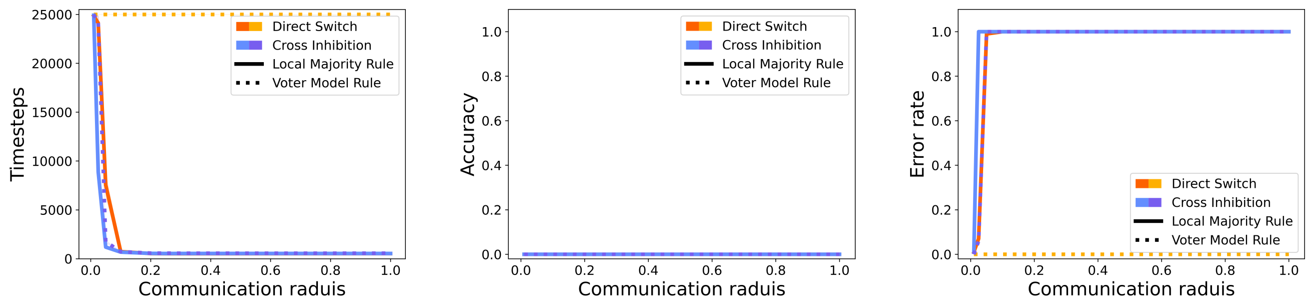
The VMR-based strategies do not manage to commit to a solution when the network comprises 15% of mad agents (figure 5.27), this is consistent with the results of the previous section: VMR-based strategies cannot handle mad agents. For LMR-based strategies with a communication radius greater or equal to 0.2 the mad agents do not have any impact as the results are similar to those obtained without them. The case where the communication radius is smaller than 0.2 has not been studied without hostile agents. Here it seems that when the communication radius is very small the agents get most if not all of their information from the environment. This leads to an error-less but very slow decision-making process as every agent must see by itself which opinion is the best. This slowness prevents the swarm from reaching a consensus in every simulation. Logically, mad agents do not impact this situation. When the agent communication radius is small, the average degree is very low, thus agents often have one or two neighbours and a LMR-based strategy will behave like a VMR-based strategy. Thus, the mad agents are very efficient at preventing commitment as they are often alone/in groups of two and can thus become a local majority. As said previously, when the communication radius increases the mad agents are inefficient as they are always a minority.



(a) Timesteps (b) Accuracy (c) Error rate

Figure 5.27: Timesteps, accuracy and error rate (proportion of agents committing to the wrong opinion) for an easy problem on a swarm with 15% of mad agents and the communication radius going from 0.01 to 1

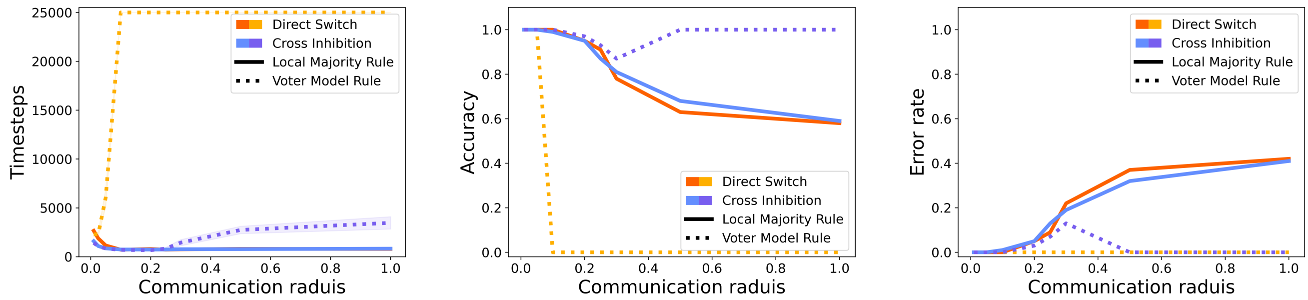
DS/VMR does not manage to commit to a solution when the network comprises 15% of zealots (figure 5.28). The other strategies can not commit when the communication radius is very low. When it increases the zealots manage to convince everyone to commit to the wrong opinion.



(a) Timesteps (b) Accuracy (c) Error rate

Figure 5.28: Timesteps, accuracy and error rate (proportion of agents committing to the wrong opinion) for an easy problem on a swarm with 15% of zealots and the communication radius going from 0.01 to 1

The figure 5.29 shows that when the communication radius is very small, the agents takes most of their information from the environment and are very accurate (but a bit slower). At this point the contrarians have no effect, indeed if there is no communication a contrarian behaves like a normal agent. For LMR-based strategies, increasing the communication radius gives a graph similar to the one obtained without hostile agents. It seems that like in the previous subsection (subsection 5.2.1) contrarians do not have an impact on LMR-based strategies. DS/VMR does not handle contrarians well. Thus, when the communication radius increases enough to for the contrarians to be able to guess which is the minority opinion (when communication radius = 0.1), the swarm stops committing and no consensus is ever reached in 25000 timesteps. DS/LMR behaves exactly like VMR-based strategies, it is unaffected until the communication radius reaches 0.25. At this point, it seems that the contrarians slow down the strategy significantly. The surprising fact is that this slowdown improves the accuracy. This can be explained by the fact that contrarians do not favour any opinions thus they only slow down the process. If the swarm is fast enough to commit within 25000 timesteps, the swarm has more time to interact with the environment and communicate more. The accuracy is thus improved. Similar results could probably be achieved by increasing the number of steps over the quorum needed to commit, the time to exit the latent states or by decreasing the probability of interacting with other agents and the environments.



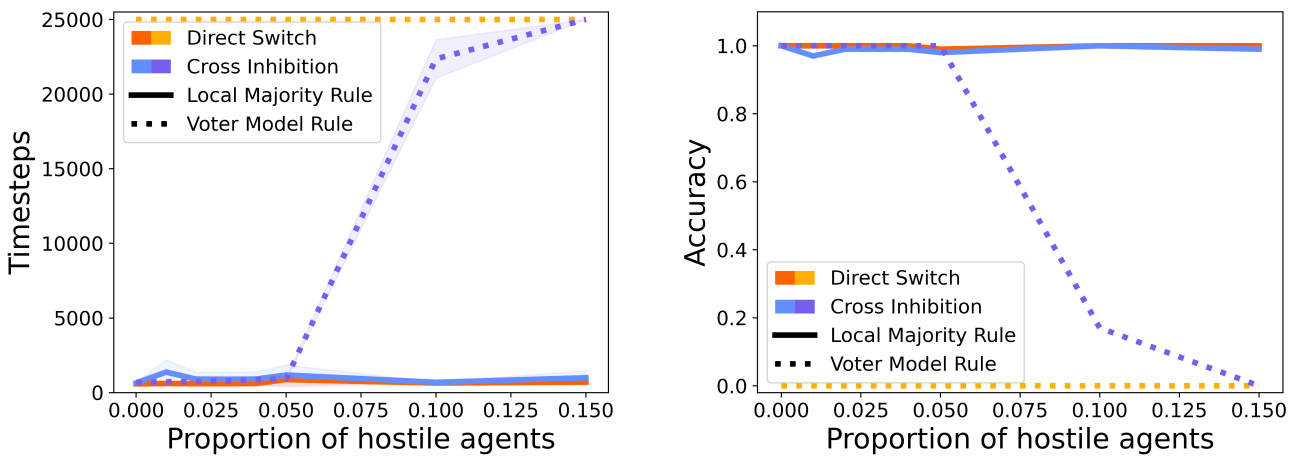
(a) Timesteps (b) Accuracy (c) Error rate

Figure 5.29: Timesteps, accuracy and error rate (proportion of agents committing to the wrong opinion) for an easy problem on a swarm with 15% of contrarians and the communication radius going from 0.01 to 1

5.2.3 Symmetry breaking with varying number of hostile agents

- VMR-based strategies are impacted by mad agents
- zealots speed-up the decision-making process, but they also ensure that their opinion is selected.
- LMR-based strategies are impacted by zealots
- contrarians are not very efficient

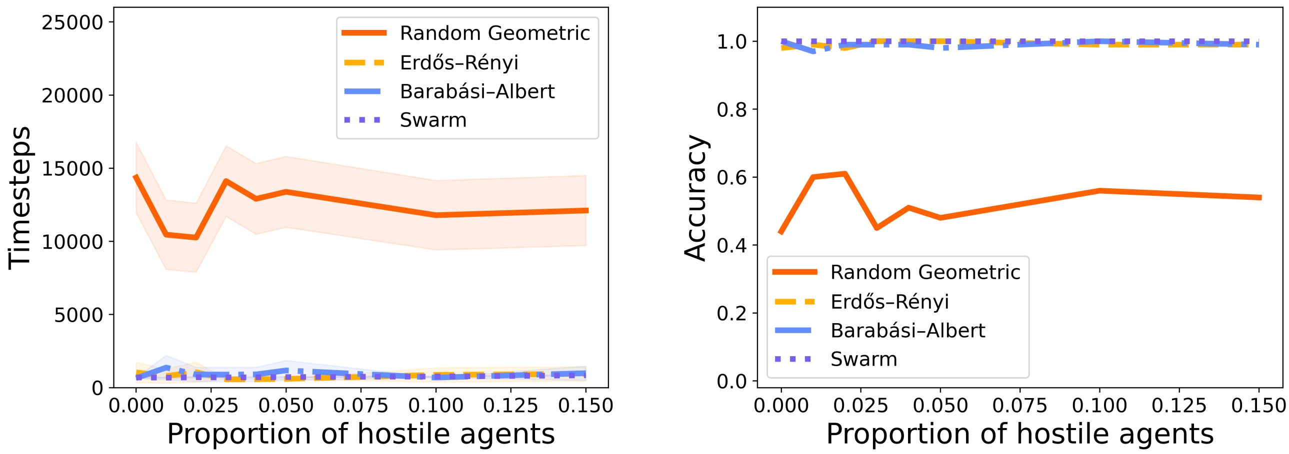
Once again, the mad agents impact strongly the VMR-based strategies by reducing the probability to reach a stable consensus and increasing the time it takes (figure 5.30). However, it does not impact the probability of choosing either opinion. The fact that LMR-based strategies are less impacted can be explained by the fact that it takes the most popular opinion over a neighbourhood. Thus, mad voter statistically cancels each other out by voting for both opinion with the same probability.



(a) Timesteps (b) Consensus probability

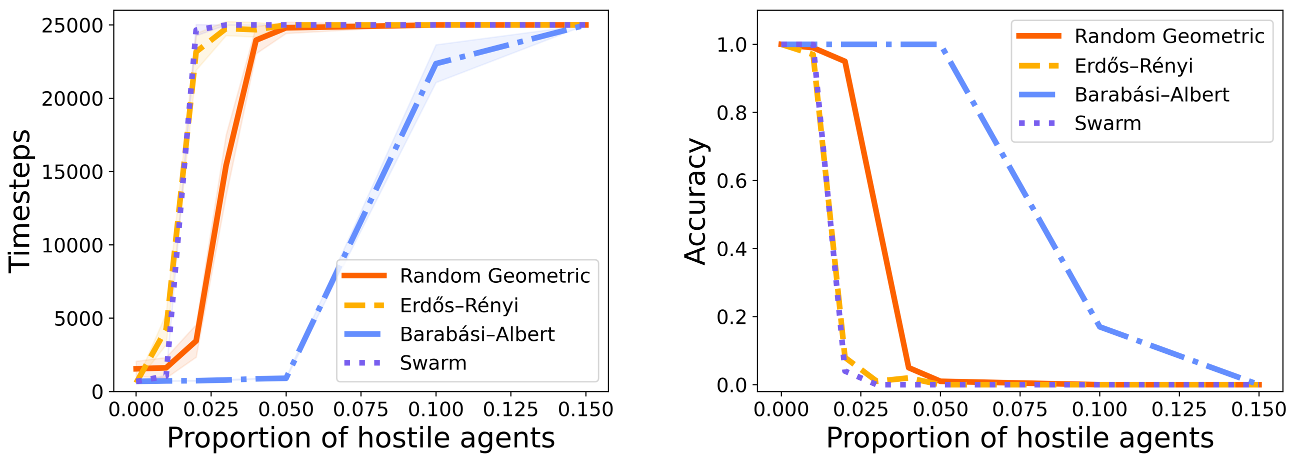
Figure 5.30: Timesteps and consensus probability for a symmetry breaking problem on a Barabási–Albert network with the proportion of mad agents going from 0 to 0.15.

Mad agents do not impact networks using LMR-based strategies (figure 5.31). However, when VMR is used, BA seems to be more resilient than other graphs (figure 5.32). The presence of agents with a lot of connections, called hubs, might help. Because those agents have a lot of neighbours, they have access to a lot of information and are very influential. Thus, they may be able to influence the swarm toward a more informed guess (and less impacted by mad agents). More research on this topic would be interesting.



(a) Timesteps (b) Consensus probability

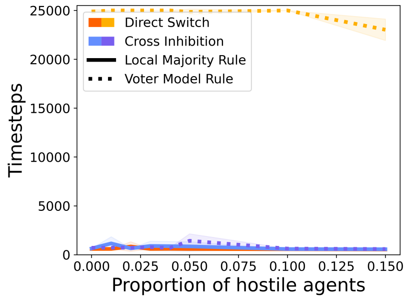
Figure 5.31: Timesteps and consensus probability for a symmetry breaking problem using cross-inhibition and local majority rule with the proportion of mad agents going from 0 to 0.15.



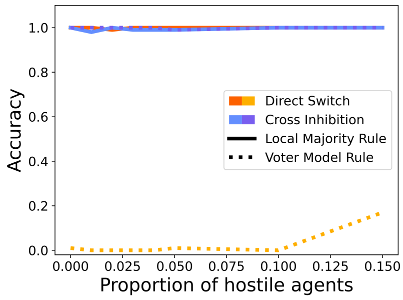
(a) Timesteps (b) Consensus probability

Figure 5.32: Timesteps and consensus probability for a symmetry breaking problem using cross-inhibition and voter model rule with the proportion of mad agents going from 0 to 0.15.

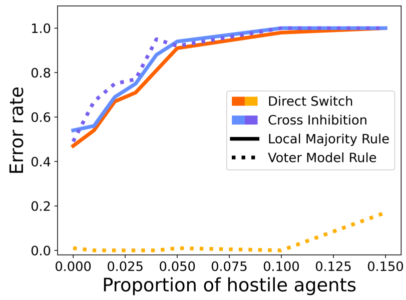
The zealots help the agents in reaching a consensus, but they also ensure that their opinion is selected.



(a) Timesteps



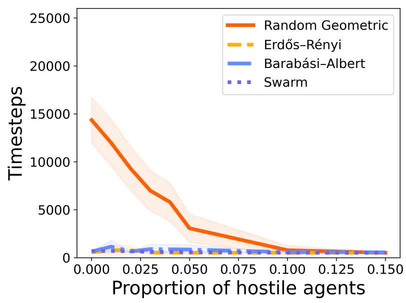
(b) Consensus probability



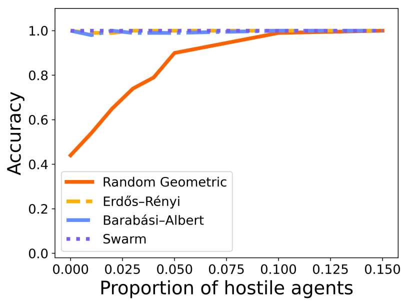
(c) proportion of agents committing to the hostile agent's opinion

Figure 5.33: Timesteps, consensus probability and proportion of agents committing to the hostile agent's opinion for a symmetry breaking problem on a Barabási–Albert network with the proportion of zealots going from 0 to 0.15.

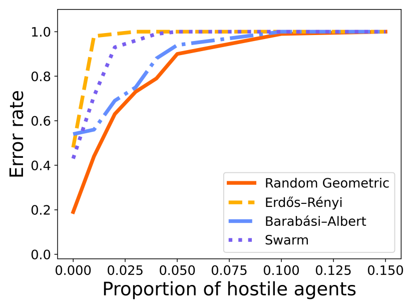
In [Galam and Jacobs, 2007] having more than 17% of zealots (using LMR) makes a side sure to win (if the opposing side does not have any zealots), here it is generally close to 10% and sometimes as low a 2% (figures 5.33 to 5.36). It is not impacted by the strategy (except DS/VMR that hardly ever manages to break the symmetry) but BA and RG seems to be better at resisting zealots. The difference between our results and [Galam and Jacobs, 2007] results can be explained by the different nature of the two experiments (need for a stable consensus, opinion can switch by interacting with the environment, network's topology, ...).



(a) Timesteps

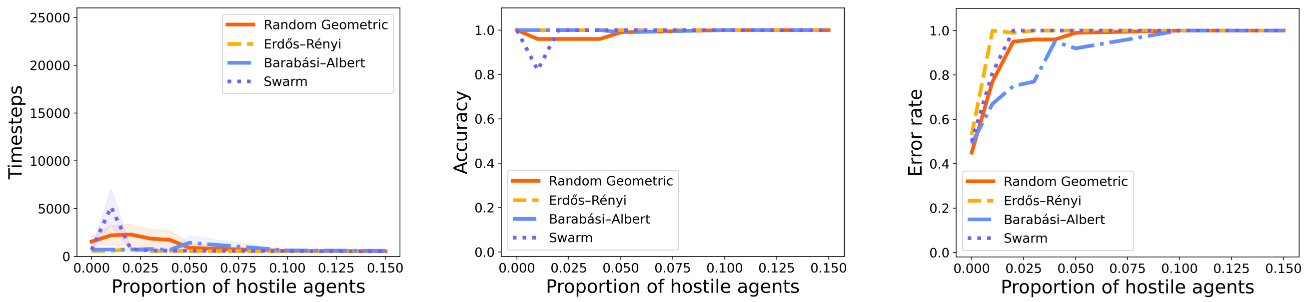


(b) Consensus probability



(c) proportion of agents committing to the hostile agent's opinion

Figure 5.34: Timesteps, consensus probability and proportion of agents committing to the hostile agent's opinion for a symmetry breaking problem using cross-inhibition and local majority rule with the proportion of zealots going from 0 to 0.15.

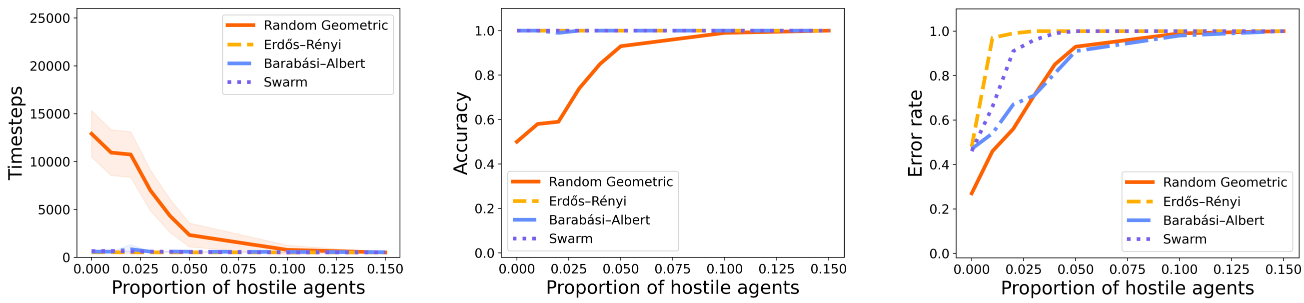


(a) Timesteps

(b) Consensus probability

(c) proportion of agents committing to the hostile agent's opinion

Figure 5.35: Timesteps, consensus probability and proportion of agents committing to the hostile agent's opinion for a symmetry breaking problem using cross-inhibition and voter model rule with the proportion of zealots going from 0 to 0.15.



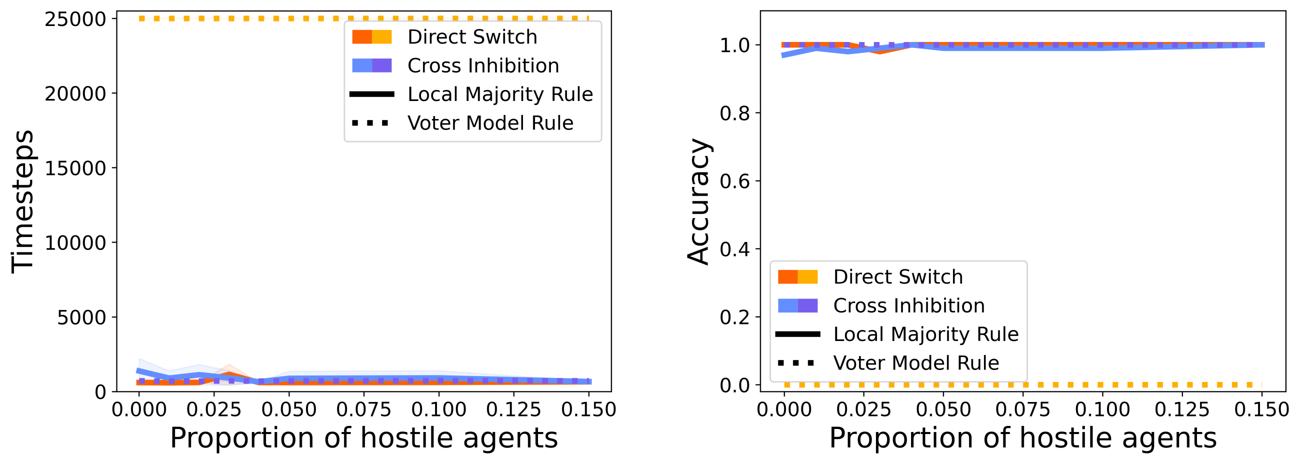
(a) Timesteps

(b) Consensus probability

(c) proportion of agents committing to the hostile agent's opinion

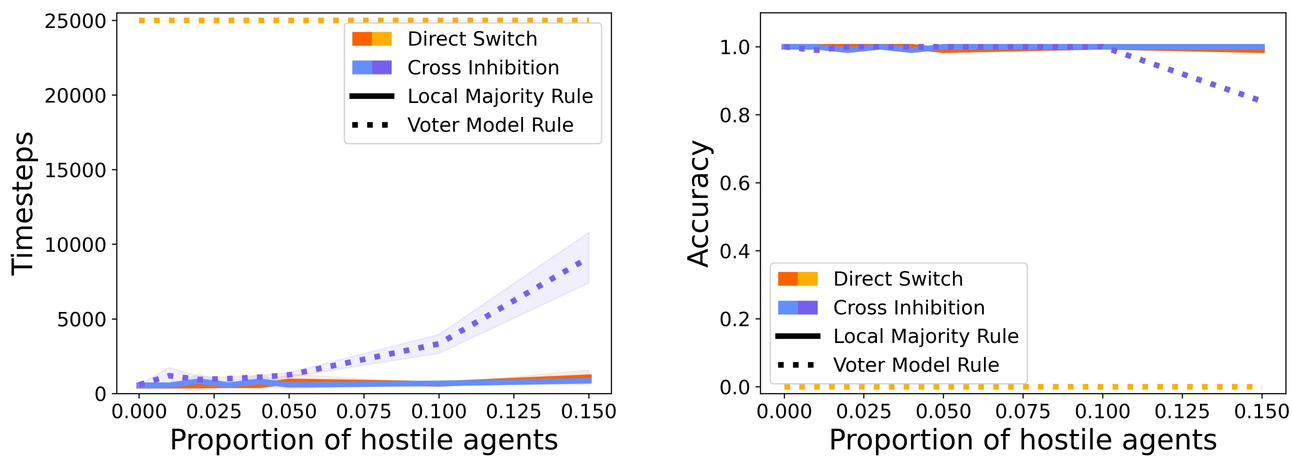
Figure 5.36: Timesteps, consensus probability and proportion of agents committing to the hostile agent's opinion for a symmetry breaking problem using using direct and local majority rule with the proportion of zealots going from 0 to 0.15.

The results (figures 5.37 to 5.41) are identical to those obtained on the easy problem (subsection 5.2.1), for majority based strategy RG was impacted while DS/VMR is only impacted on the ER networks if there is more than 5% of contrarian.



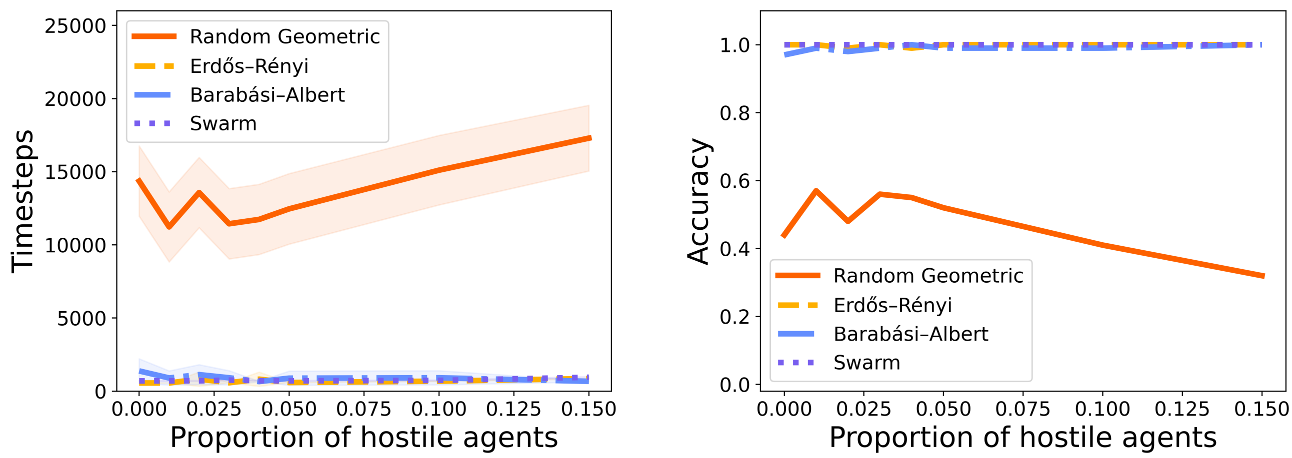
(a) Timesteps (b) Consensus probability

Figure 5.37: Timesteps and consensus probability for a symmetry breaking problem on a Barabási–Albert network with the proportion of contrarians going from 0 to 0.15.



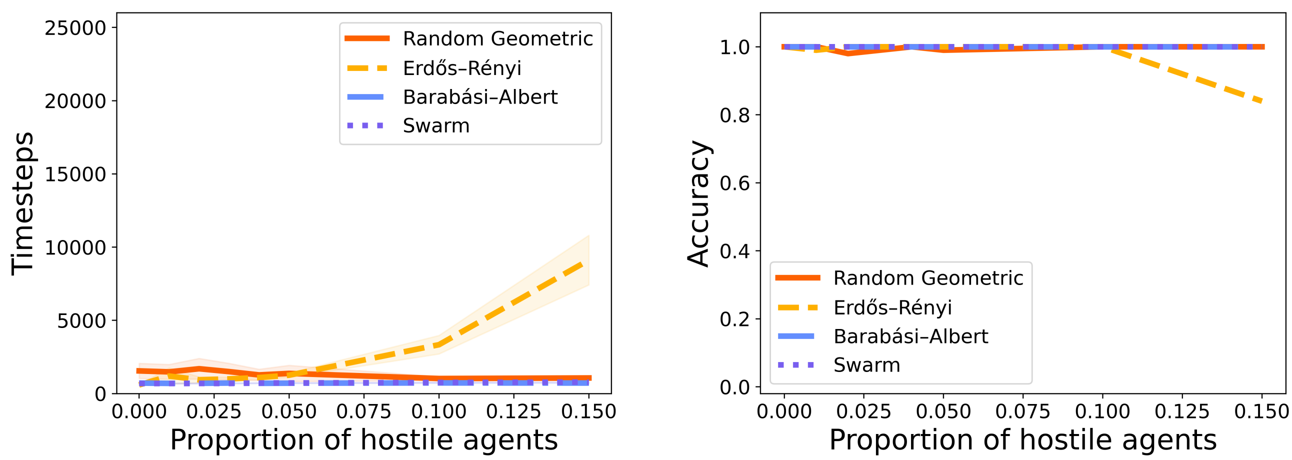
(a) Timesteps (b) Consensus probability

Figure 5.38: Timesteps, consensus probability and proportion of agents committing to the hostile agent's opinion for a symmetry breaking problem on a Erdos–Rényi network with the proportion of contrarians going from 0 to 0.15.



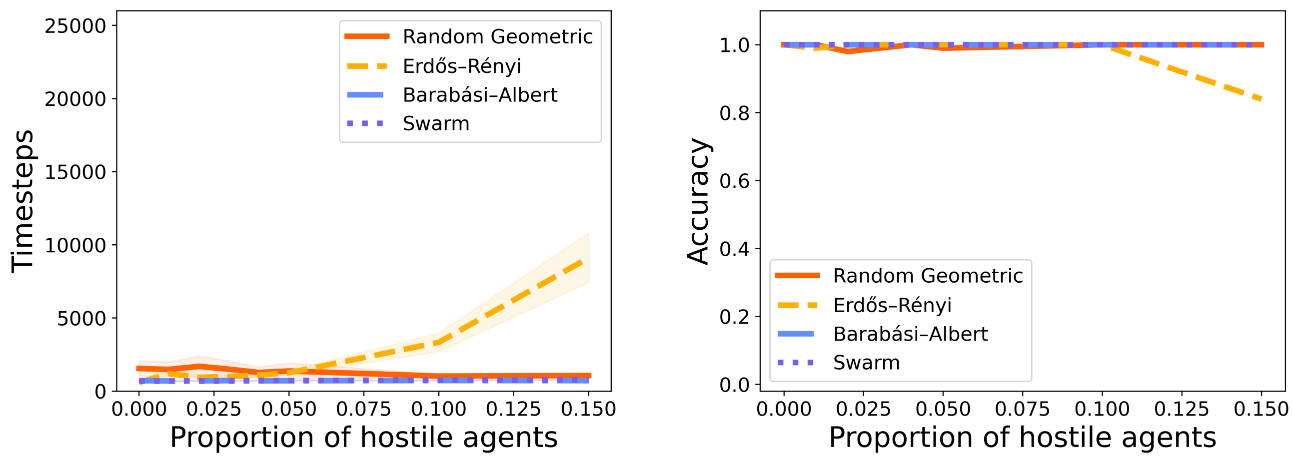
(a) Timesteps (b) Consensus probability

Figure 5.39: Timesteps and consensus probability for a symmetry breaking problem using cross-inhibition and local majority rule with the proportion of contrarians going from 0 to 0.15.



(a) Timesteps (b) Consensus probability

Figure 5.40: Timesteps and consensus probability for a symmetry breaking problem using cross-inhibition and voter model rule with the proportion of contrarians going from 0 to 0.15.



(a) Timesteps

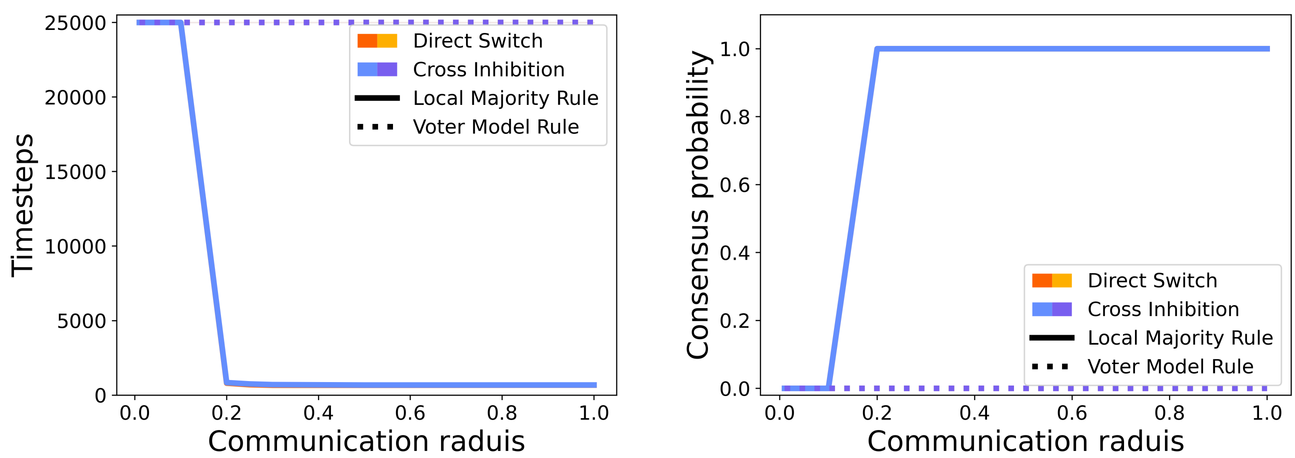
(b) Consensus probability

Figure 5.41: Timesteps and consensus probability for a symmetry breaking problem using using direct and local majority rule with the proportion of contrarians going from 0 to 0.15.

5.2.4 Symmetry breaking on swarm with varying agent's communication radius

- if the communication radius is high enough, LMR-based strategies ignore mad agents
- contrarians and mad agents can prevent swarm using the VMR-based strategies from reaching a consensus

When the communication radius is low, the agents do not manage to break the symmetry (figure 5.42). This is expected as they receive most of their information from the environment. However, when it increases the LMR-based agents managed to break it quickly. This is probably due to the fact that, when a consensus is reached, a mad agent that suddenly switches its opinion does not convince anyone else to switch (it is a minority). Thus, the consensus can be maintained during 500 steps. Instead, the VMR-based strategies oscillate between a consensus on the first option and a consensus on the second. Once again, the mad agents do not appear to have any effect on VMR-based strategies.

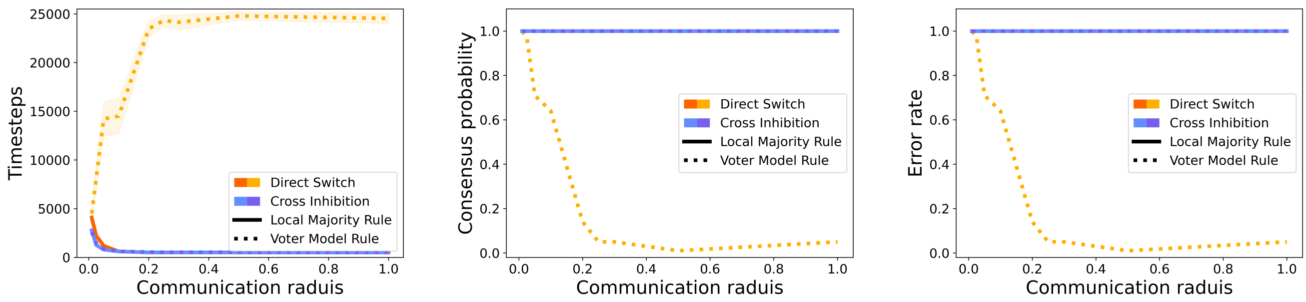


(a) Timesteps

(b) Accuracy

Figure 5.42: Timesteps and consensus probability for an symmetry breaking problem on a swarm with 15% of mad agents and the communication radius going from 0.01 to 1.

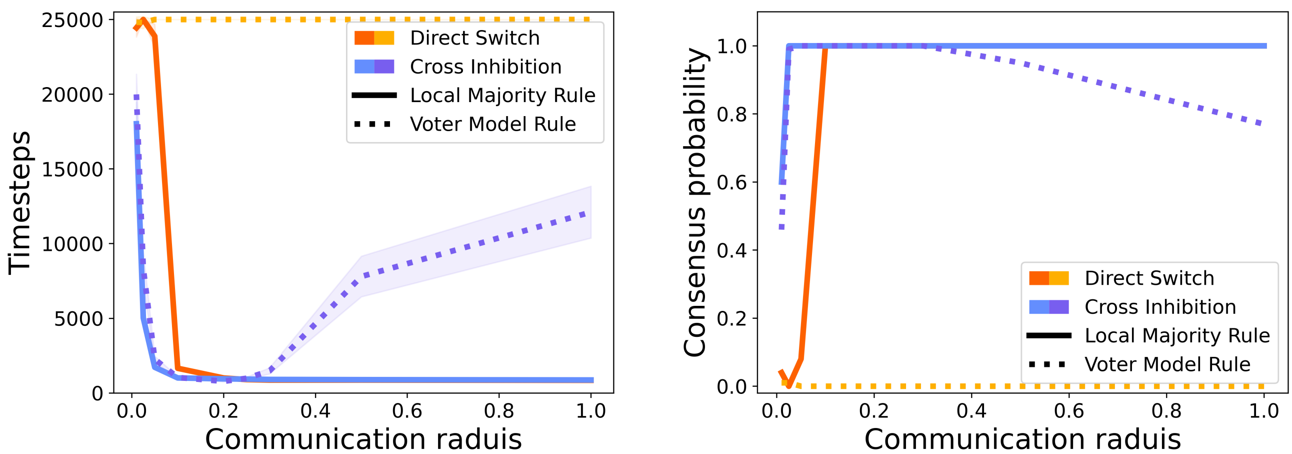
When the communication radius is very small, the agents are mainly interacting with the environments but, after a few thousands steps they still all break the symmetry siding with the zealots opinion (figure 5.43) like in subsection 5.1.2. When the communication radius and the average degree increases. It makes the problem much harder for DS/LMR. The three other strategies just commit faster to the zealots opinions.



(a) Timesteps (b) Accuracy (c) Error rate

Figure 5.43: Timesteps, consensus probability and proportion of agents committing to the hostile agent's opinion for an symmetry breaking problem on a swarm with 15% of zealots and the communication radius going from 0.01 to 1.

DS/LMR newer breaks the symmetry (figure 5.44). The other strategies results depend on the communication radius. When the communication radius is very small, DS/VMR rarely breaks the symmetry. CI-based strategies do it, 60% of the times. It is normal that they have a hard time breaking the symmetry as they do not have the opportunity to communicate a lot. They get better results than the swarm network attacked by mad agents because contrarians behave like normal agents when they do not have any neighbours to communicate with. When the communication radius is small those three strategies manage to break the symmetry quickly and consistently. However, when it increases only the CI based strategies continue to always break the symmetry while DS/VMR takes more time and only succeeds 80% of the times. This slowdown is also observed in an easy problem in subsection 5.2.2 where it increases the accuracy. Here it is just a drawback as we do not care what opinion is chosen (the contrarians do not impact it). Moreover, it slows down some simulations enough to prevent them from reaching a stable consensus within 25000 steps, thus decreasing accuracy.



(a) Timesteps (b) Accuracy

Figure 5.44: Timesteps, consensus probability for an symmetry breaking problem on a swarm with 15% of contrarians and the communication radius going from 0.01 to 1.

Chapter 6

Discussion & Conclusion

6.1 Comparison of the four methods in a fully cooperative environment

When working in a fully cooperative environments, strategies seems to have a greater impact than the networks topology. Of all the networks, random geometric is the only one that seems to always get inferior results, while none of the other networks is the best in every situation. An important parameter that was not tested is the average degree, if the swarm network had the same average degree as RG (and better results) BA and ER had much lower average degree. Doing the same experiment with the same average degree on all networks could be interesting (even if it the average degree would be high). Overall, it still seems that networks should thus be random or time varying (the connections are changed frequently). One of the strategies also had inferior results, indeed DS/VMR cannot handle symmetry breaking problems (this was already established in [Reina et al., 2017]) and is far slower than all the others strategies. Thus, its use should be avoided. Its only interesting property is that in best-of-2 problems it does not commit to an inferior solution (it never makes any mistake). However, it can only solve in a reasonable time very easy problems where also others strategies have extremely good accuracy. Another interesting point is that it seems that LMR based strategies do not handle highly connected networks well in best-of-2 problem, this might be due to the fact that when the majority of the agents commit to an inferior opinion, it is difficult to switch to the other opinion. However, LMR based strategies performances increase with the network's average degree in symmetry breaking problems. A surprising fact that might be interesting for future research is understanding why and how CI/VMR performances in symmetry breaking problems with small communication radius can outperform the other strategies.

When working on a best-of-n problem with $n > 2$ (and one opinion with a superior quality), the more options an agent has, the more likely it selects the wrong one. Moreover, at each step the agent has a 20% chance of estimating the quality value of a random option. Thus, each time we add an option, we decrease the probability of selecting the right one. This decreases both the accuracy and the time taken to reach a stable consensus.

6.2 Comparison of the four methods in a non-cooperative environment

It is clear that zealots are the most efficient hostile agents, as they works on all strategies. Indeed if the communication radius is not very small 15 zealots in a swarm of 100 agents will manage to make sure that their opinion is picked or prevent the swarm from reaching a consensus in 25000

timesteps. Mad agents are extremely efficient on VMR based strategies, but if the average degree is high enough, they are useless on LMR based strategies. Contrarians are the worst of the three tested non-cooperative agents because they also only impact VMR based strategies but have a weaker effect (especially on CI/VMR).
CI/LMR and DS/LMR have very similar behaviour, they are insensible to contrarians and mad agents but are sensible to zealots. CI/VMR has a behaviour similar to LMR based strategies when attacked by zealots, however it is sensible to large mad agents and large amount of contrarians. An interesting fact is that adding contrarians to a swarm following CI/VMR in a best-of-n problem slows it down but, if the problem is easy enough to still be solved in less than 25000 steps, this slow down improves accuracy. Future research could investigate the benefits of slowing down swarms in certain cases. These results also suggest a reconsideration of the role of contrarians in our societies, does their action force us to debate subjects that would otherwise not have been debated ? The last strategy, DS/VMR is not able to achieve a consensus when a few hostile agents are introduced. However, it is the only strategy that, when attacked by zealots, does not commit to their opinions. Regarding the networks analysis, BA seems to be the most resilient while ER seems to have the worst results. Swarm and RG generally have a similar accuracy, but swarm tend to reach a consensus faster.

Bibliography

[Barabasi and Albert, 1999] Barabasi, A.-L. and Albert, R. (1999). Emergence of scaling in random networks. *Science*, 286(5439):509–512.

[Baumann et al., 2020] Baumann, F., Lorenz-Spreen, P., Sokolov, I. M., and Starnini, M. (2020). Modeling echo chambers and polarization dynamics in social networks. *Phys. Rev. Lett.*, 124:048301.

[Biancalani et al., 2014] Biancalani, T., Dyson, L., and McKane, A. J. (2014). Noise-induced bistable states and their mean switching time in foraging colonies. *Phys. Rev. Lett.*, 112:038101.

[Canciani et al., 2019] Canciani, F., Talamali, M. S., Marshall, J. A. R., and Reina, A. (2019). Keep calm and vote on: Swarm resiliency in collective decision making. Technical report.

[Castellano et al., 2009] Castellano, C., Fortunato, S., and Loreto, V. (2009). Statistical physics of social dynamics. *Rev. Mod. Phys.*, 81:591–646.

[Cinelli et al., 2021] Cinelli, M., Morales, G. D. F., Galeazzi, A., Quattrociocchi, W., and Starnini, M. (2021). The echo chamber effect on social media. *Proceedings of the National Academy of Sciences*, 118(9):e2023301118.

[Cinus et al., 2021] Cinus, F., Minici, M., Monti, C., and Bonchi, F. (2021). The effect of people recommenders on echo chambers and polarization.

[Clifford and Sudbury, 1973] Clifford, P. and Sudbury, A. (1973). A model for spatial conflict. *Biometrika*, 60(3):581–588.

[de Arruda et al., 2021] de Arruda, H. F., Benatti, A., Silva, F. N., Comin, C. H., and da Fontoura Costa, L. (2021). Contrarian effects and echo chamber formation in opinion dynamics. *Journal of Physics: Complexity*, 2(2):025010.

[De Masi et al., 2021] De Masi, G., Prasetyo, J., Zakir, R., Mankovskii, N., Ferrante, E., and Tuci, E. (2021). Robot swarm democracy: the importance of informed individuals against zealots. *Swarm Intelligence*, 15.

[Euler, 1741] Euler, L. (1741). Solutio problematis ad geometriam situs pertinentis. *Commentarii academiae scientiarum Petropolitanae*, 8:128–140.

[Fienberg, 2012] Fienberg, S. E. (2012). A brief history of statistical models for network analysis and open challenges. *Journal of Computational and Graphical Statistics*, 21(4):825–839.

[Galam, 2004] Galam, S. (2004). Contrarian deterministic effects on opinion dynamics: "the hung elections scenario". *Physica A: Statistical Mechanics and its Applications*, 333:453–460.

[Galam et al., 2002] Galam, S., Chopard, B., and Droz, M. (2002). Killer geometries in competing species dynamics. *Physica A: Statistical Mechanics and its Applications*, 314.

[Galam and Jacobs, 2007] Galam, S. and Jacobs, F. (2007). The role of inflexible minorities in the breaking of democratic opinion dynamics. *Physica A: Statistical Mechanics and its Applications*, 381:366–376.

[Galam, S., 2002] Galam, S. (2002). Minority opinion spreading in random geometry. *Eur. Phys. J. B*, 25(4):403–406.

[Gekle et al., 2005] Gekle, S., Peliti, L., and Galam, S. (2005). Opinion dynamics in a three-choice system. *The European Physical Journal B*, 45(4):569–575.

[Gilbert, 1959] Gilbert, E. N. (1959). Random Graphs. *The Annals of Mathematical Statistics*, 30(4):1141 – 1144.

[Hagberg et al., 2008] Hagberg, A. A., Schult, D. A., and Swart, P. J. (2008). Exploring network structure, dynamics, and function using networkx. In Varoquaux, G., Vaught, T., and Millman, J., editors, *Proceedings of the 7th Python in Science Conference*, pages 11 – 15, Pasadena, CA USA.

[Holley and Liggett, 1975] Holley, R. A. and Liggett, T. M. (1975). Ergodic Theorems for Weakly Interacting Infinite Systems and the Voter Model. *The Annals of Probability*, 3(4):643 – 663.

[Krapivsky and Redner, 2003] Krapivsky, P. L. and Redner, S. (2003). Dynamics of majority rule in two-state interacting spin systems. *Phys. Rev. Lett.*, 90:238701.

[Lerman and Galstyan, 2002] Lerman, K. and Galstyan, A. (2002). Mathematical model of foraging in a group of robots: Effect of interference. *Autonomous Robots*, 13(2):127–141.

[Mobilia, 2003] Mobilia, M. (2003). Does a single zealot affect an infinite group of voters? *Phys. Rev. Lett.*, 91:028701.

[Mobilia and Georgiev, 2005] Mobilia, M. and Georgiev, I. T. (2005). Voting and catalytic processes with inhomogeneities. *Phys. Rev. E*, 71:046102.

[Mobilia et al., 2007] Mobilia, M., Petersen, A., and Redner, S. (2007). On the role of zealotry in the voter model. *Journal of Statistical Mechanics: Theory and Experiment*, 2007(08):P08029–P08029.

[Pais et al., 2013] Pais, D., Hogan, P. M., Schlegel, T., Franks, N. R., Leonard, N. E., and Marshall, J. A. R. (2013). A mechanism for value-sensitive decision-making. *PLOS ONE*, 8(9):1–9.

[Pirrone et al., 2022] Pirrone, A., Reina, A., Stafford, T., Marshall, J. A., and Gobet, F. (2022). Magnitude-sensitivity: rethinking decision-making. *Trends in Cognitive Sciences*, 26(1):66–80.

[Prasetyo et al., 2020] Prasetyo, J., De Masi, G., Tuci, E., and Ferrante, E. (2020). *The Effect of Differential Quality and Differential Zealotry in the Best-of-¡i¿n¡/i¿ Problem*, page 65–66. Association for Computing Machinery, New York, NY, USA.

[Redner, 2019] Redner, S. (2019). Reality-inspired voter models: A mini-review. *Comptes Rendus Physique*, 20(4):275–292.

[Reina et al., 2018] Reina, A., Bose, T., Trianni, V., and Marshall, J. A. R. (2018). *Effects of Spatiality on Value-Sensitive Decisions Made by Robot Swarms*, pages 461–473. Springer International Publishing, Cham.

[Reina et al., 2017] Reina, A., Marshall, J. A. R., Trianni, V., and Bose, T. (2017). Model of the best-of-n nest-site selection process in honeybees. *Phys. Rev. E*, 95:052411.

[Reina et al., 2015a] Reina, A., Miletitch, R., Dorigo, M., and Trianni, V. (2015a). A quantitative micro-macro link for collective decision: the shortest path discovery/selection example. *Swarm Intelligence*.

[Reina et al., 2015b] Reina, A., Valentini, G., Fernández-Oto, C., Dorigo, M., and Trianni, V. (2015b). A design pattern for decentralised decision making. *PLOS ONE*, 10(10):1–18.

[Seeley et al., 2012] Seeley, T. D., Visscher, P. K., Schlegel, T., Hogan, P. M., Franks, N. R., and Marshall, J. A. R. (2012). Stop signals provide cross inhibition in collective decision-making by honeybee swarms. *Science*, 335(6064):108–111.

[Sood et al., 2008] Sood, V., Antal, T., and Redner, S. (2008). Voter models on heterogeneous networks. *Physical Review E*, 77(4).

[Sood and Redner, 2005] Sood, V. and Redner, S. (2005). Voter model on heterogeneous graphs. *Physical Review Letters*, 94(17).

[Starnini et al., 2016] Starnini, M., Frasca, M., and Baronchelli, A. (2016). Emergence of metapopulations and echo chambers in mobile agents. *Scientific Reports*, 6(1).

[Stauffer, 2002] Stauffer, D. (2002). Percolation and galam theory of minority opinion spreading. *International Journal of Modern Physics C*, 13:975–977.

[Stauffer and Martins, 2003] Stauffer, D. and Martins, J. S. S. (2003). Simulation of galam's contrarian opinions.

[Talamali et al., 2019] Talamali, M. S., Marshall, J. A. R., Bose, T., and Reina, A. (2019). Improving collective decision accuracy via time-varying cross-inhibition. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 9652–9659.

[Talamali et al., 2021] Talamali, M. S., Saha, A., Marshall, J. A. R., and Reina, A. (2021). When less is more: Robot swarms adapt better to changes with constrained communication. *Science Robotics*, 6(56):eabf1416.

[Tessone et al., 2004] Tessone, C. J., Toral, R., Amengual, P., Wio, H. S., and San Miguel, M. (2004). Neighborhood models of minority opinion spreading. *The European Physical Journal B*, 39(4):535–544.

[Trianni et al., 2016] Trianni, V., De Simone, D., Reina, A., and Baronchelli, A. (2016). Emergence of consensus in a multi-robot network: From abstract models to empirical validation. *IEEE Robotics and Automation Letters*, 1(1):348–353.

[University of Sheffield, 2020] University of Sheffield (2020). Demamas. `https://github.com/DiODeProject/DeMaMAS`.

[Valentini et al., 2016a] Valentini, G., Brambilla, D., Hamann, H., and Dorigo, M. (2016a). Collective perception of environmental features in a robot swarm. In Dorigo, M., Birattari, M., Li, X., López-Ibáñez, M., Ohkura, K., Pinciroli, C., and Stützle, T., editors, *Swarm Intelligence*, pages 65–76, Cham. Springer International Publishing.

[Valentini et al., 2017] Valentini, G., Ferrante, E., and Dorigo, M. (2017). The best-of-n problem in robot swarms: Formalization, state of the art, and novel perspectives. *Frontiers in Robotics and AI*, 4.

[Valentini et al., 2016b] Valentini, G., Ferrante, E., Hamann, H., and Dorigo, M. (2016b). Collective decision with 100 kilobots: Speed versus accuracy in binary discrimination problems. *Autonomous Agents and Multi-Agent Systems*, 30(3):553–580.

[Valentini et al., 2014] Valentini, G., Hamann, H., and Dorigo, M. (2014). Self-organized collective decision making: The weighted voter model. volume 1.

[Valentini et al., 2015] Valentini, G., Hamann, H., and Dorigo, M. (2015). Efficient decision-making in a self-organizing robot swarm: On the speed versus accuracy trade-off. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '15, page 1305–1314, Richland, SC. International Foundation for Autonomous Agents and Multiagent Systems.

[Xie et al., 2011] Xie, J., Sreenivasan, S., Korniss, G., Zhang, W., Lim, C., and Szymanski, B. (2011). Social consensus through the influence of committed minorities. *Physical Review E*, 84:011130.

[Zhang et al., 2014] Zhang, W., Lim, C. C., Korniss, G., and Szymanski, B. K. (2014). Opinion dynamics and influencing on random geometric graphs. *Scientific Reports*, 4(1).

Appendix A

List of parameters

This appendix provide the full list of parameters used to run DeMaMAS, the full explanation of each parameters can be found in the README.md on Github (NetworkX branch) [University of Sheffield
General parameters:

COMPOSITION=1

ENVIRONMENT_SIZE=1

NUM_SIMULATION=100

NUM_STEP=25000

NUM_TOTEM=[1]

OVERLAPPING_TOTEMS="false"

QUORUM=0.8

REQUIRED_NUMBER_OF_STEP_OVER_Q=500

STARTING_POINT="false"

INITIAL_OPINIONS=[0]

COMPOSITION=[1]

Agent and totem characteristics parameters:

TOTEM_RADIUS=0

FIRST_ENTRY_ONLY_DISC="false"

Movement parameters (only for swarm):

MOVE_DIMENSION=0.005

STRAIGHT_LENGTH=20

Behaviour parameters:

DISCOVERY_METHOD=[probabilistic]

PROBABILISTIC_DISCOVERY_PROPORTION=[difference]

K_UNANIMITY_PARAMETER=2

MAX_QUALITY=1

ZEALOT_QUALITY=[0.5]

Message parameters:

MSG_TYPE=simple

SEND_METHOD=[wv]

SEND_CONSTANT_INTERVAL=1

FILTER_MSG_PARAM=[0]

Decay parameters:

DECAY_METHOD=[constant]

DECAY_STRENGTH=[0] this means that there is no decay ie: the agent never forget its own opinion

Interaction function parameters :

PRE_STEP_SELF_STRENGTH=[0.2]

PRE_STEP_SOCIAL_STRENGTH=[1]

INTERACTIVE_PROBABILITY=0.05

INTERACTION_FUNCTION=[constant] thus the post_X_STRENGTH are never used as the interaction rate between agents is never updated

Changing parameters, each of them as one or more base values and a set of values tested using the other parameters base values:

NUMBER_OF_EDGES=3 (2 3 5) # for BA

EDGE_PROBABILITY=0.03 (0.04 0.05 0.1 0.5 1) # for ER

AGENT_RADIUS=0.2 (0.25 0.3 0.5 1) # for RG and swarm

QUALITY_VALUES=([0.4,0.8] [0.72,0.8] [0.5,0.5]) ([0.5,0.8] ([0.6,0.8] [0.7,0.8] [0.725,0.8] [0.75,0.8] [0.775,0.8])

NUMBER_OF_OPINIONS=(2) (2 3 5 10 15) with QUALITY_VALUES being [0.4,... 0.8] (NUMBER_OF_OPINIONS-1 time 0.4, and 0.8)

DEVIATION_VALUE= 1 (0 0.2 0.5)

NUMBER_OF_AGENTS= 100 (150 200 250 500) sometimes up to 2000

UPDATE_MODEL=([direct] [crossInhibition])

UPDATE_RULE=([majority] [random])