

## Modulating Interaction Times in an Artificial Society of Robots

Yara Khaluf<sup>1</sup> and Heiko Hamann<sup>2</sup>

<sup>1</sup>IDLab, Ghent University - imec, Belgium

<sup>2</sup>Institute of Computer Engineering, University of Lübeck, Germany  
yara.khaluf@ugent.be

### Abstract

In a collaborative society, sharing information is advantageous for each individual as well as for the whole community. Maximizing the number of agent-to-agent interactions per time becomes an appealing behavior due to fast information spreading that maximizes the overall amount of shared information. However, if malicious agents are part of society, then the risk of interacting with one of them increases with an increasing number of interactions. In this paper, we investigate the roles of interaction rates and times (aka edge life) in artificial societies of simulated robot swarms. We adapt their social networks to form proper trust sub-networks and to contain attackers. Instead of sophisticated algorithms to build and administrate trust networks, we focus on simple control algorithms that locally adapt interaction times by changing only the robots' motion patterns. We successfully validate these algorithms in collective decision-making showing improved time to convergence and energy-efficient motion patterns, besides impeding the spread of undesired opinions.

### Introduction

Human societies are built upon social networks that form the infrastructure for sharing information, such as ideas (Leskovec et al., 2009; Orellana-Rodriguez and Keane, 2018), product innovations (Leskovec et al., 2007; Zhong et al., 2018), and political movements (Polletta and Jasper, 2001). The network metaphor suggests a discrete and static property (edge/no edge), however, agent-to-agent interactions have certainly different intensities and durations. Specific characteristics of these agent-to-agent interactions influence global phenomena, such as consensus formation in collective decision-making processes or the spread of an epidemic (Keeling and Rohani, 2011; Heesterbeek et al., 2015).

The interactions of an individual are key to understand information spreading processes in its neighborhood. They can significantly influence its opinions and define its information spreading capability. Local neighborhoods are relevant, as shown for example by Christakis and Fowler (2007) who report that your chances of being obese are probabilistically characterized by your social network. In turn, obesity as a global feature of society can also be analyzed based on the social interactions among individuals.

Furthermore, dynamic individual interactions drive the emergence of communities (Hess et al., 2016). In general, social systems have features that develop on different time scales ranging from micro-seconds to years. Spreading information travels magnitudes faster than the speed of change of the underlying network topology. Hence, in simplified models the topology can be approximated by static networks (Krings et al., 2012). Models representing interaction times may, however, prove to be more powerful to study the spread of information.

Here, we study the dynamics of information spreading in large societies as observed in artificial and natural collective systems. We focus on the timescales of topology dynamics (i.e., community formation) and their impact on global features. As key parameter of our study we choose the average and distribution of agent-to-agent interaction times. As interaction time we define the uninterrupted time spent by two agents in mutual communication range. The distribution of the interaction time is also known as edge life distribution (Yang et al., 2013). Our main inspiration comes from Meier (1962) who reports the importance of decreased interaction times in modern cities Khaluf (2017). We study two complementary agent behaviors that either (a) share a desirable piece of information or (b) trap an undesired piece of information. In natural systems, such as collective decision-making in ants (Pratt et al., 2002), trust is virtually taken for granted and spreading of information maximized. However, ants tweak their social network when they need to fight an epidemic (Stroeymeyt et al., 2018). In artificial systems, such as collective decision-making in swarm robots (Khaluf et al., 2018; Hamann et al., 2014; Valentini et al., 2016; Rausch et al., 2018), sharing of information is to be maximized and in many studies trust is also virtually taken for granted. Only few works on decentralized error detection focus on identifying and excluding damaged robots (Lau et al., 2011; Tarapore et al., 2015). In our study, we assume that not all information is useful but possibly harmful. We extend the current state of the art in collective decision-making of artificial societies by considering how to deal with undesired information. The focus is on how to modulate agent-to-agent interaction times

to either boost or impede spread of information.

We investigate our hypothesis using an artificial collective system, a simulated robot swarm (Hamann, 2018b), that has a dynamic topology. By using a collective robot system we restrict ourselves to local links of directly interacting neighbors. This spatial restriction allows us to draw relations to models of human networks for different spatial densities (e.g., urban vs. rural). Furthermore, simulated robot swarms serve here as systems that are easy to track and to engineer. We tune the interaction dynamics of the swarm (e.g., interaction times) to achieve desired collective behaviors, such as a decision consensus in collective decision-making. We examine the role of interaction times in a symmetry-breaking problem, that is, agents need to select between two options of equal quality. Two parameters are important to optimize: (1) degree of coherence, the percentage of robots that finally agree on the same option, in the best case consensus (100% agreement) and (2) time to convergence, the time to achieve a stable degree of coherence. The time to converge is particularly important in the symmetry-breaking type of decision-making, because both options are of the same quality, hence, further gathering of information is wasted time and cannot increase the accuracy of the decision. As commonly assumed in collective decision-making scenarios, we have an instantaneous exchange of information (opinions) between neighboring agents. We analyze the role of the interaction time under two modes (1) offline tuning of interaction times by adapting the population density externally (Khaluf et al., 2017), and (2) active tuning of interaction times by a set of proposed algorithms that exploit robot motion patterns to modulate the distribution of the agent-to-agent interaction times. The physical interaction time was highlighted as a key parameter in the emergence of collective motion behaviors by Stark et al. (2008).

### Approach to modulate interaction times

We use a homogeneous swarm of  $N$  simulated robots.<sup>1</sup> Robots wander randomly in their arena and while moving they interact with their local neighborhood that is defined based on their communication range. All robots within communication range are neighbors and interactions are mutual. Therefore, the robot control algorithm influences the distribution of agent-to-agent interaction times but also the spatial distribution of robot density. (number of robots per area).

We consider a symmetry-breaking problem (Hamann et al., 2012). A binary collective decision-making problem with options  $A$  and  $B$  of the same quality. The robot swarm is asked to achieve a consensus on either one of them. This is a well-studied problem known from different fields, such as physics, biology, opinion dynamics, and others.

### Study setups

Our study consists of the following experiment setups:

---

<sup>1</sup>Footbot, [http://www.swarmanoid.org/swarmanoid\\_hardware.php](http://www.swarmanoid.org/swarmanoid_hardware.php)

### A - interaction time as a function of population density:

In preliminary experiments, we measure and analyze the dependency of agent-to-agent interaction time changes on swarm density. We explore possibilities of tweaking interaction times offline by choosing proper swarm densities.

### B - boosting spread of desired information in a decision-making system with majority rule:

We apply the majority rule in a symmetry-breaking scenario, the spreading information is the robots' opinions, that are in this case desired to spread widely. We define two sets of experiments here. First, we use offline tuning of interaction times (i.e., swarm density). Second, we validate our control algorithm to actively tune interaction times in order to achieve a system with an enhanced well-mixed property (i.e., closer to the ideal of a well-mixed system where the chance of any agent to interact with any other agent is equal). We modify the robot motion to improve the diffusion of robots. The result are decreased agent-to-agent interaction times and an increased chance of encountering more neighbors, that increase a robot's opinion sample. Well-mixed populations are advantageous in collective decision-making. However, once consensus is reached the robots may not need to continue their costly motion pattern. We propose an energy-aware algorithm that creates a global self-awareness of the degree of coherence in the swarm. Each robot samples opinions of its neighborhood and measure the degree of coherence. In the case of high coherence over a long-enough period, robots slow down until they stop and preserve their current neighborhood as it is not necessary for the decision-making process to move further.

### C - impeding spread of undesired information in a decision-making system with majority rule:

We consider a heterogeneous swarm with two sub-populations. One sub-population uses the majority rule as above. The other sub-population is assumed to be malicious. They contradict the majority in their neighborhood and adopt the opposite opinion. In the literature they are called contrarians (Gambaro and Crokidakis, 2017; Khalil and Toral, 2019; Hamann, 2018a). First, we show that contrarians reduce coherence. Second, we show how our proposed control algorithm can tune interaction times with contrarians and contain them. Their influence is then limited and the degree of coherence is increased. We highlight that extending agent-to-agent interaction times can be of a significant benefit in this setup. This is in contrast to standard setups in collective decision-making where short interaction times are preferred because they improve the mixing and spread of opinions.

### Motion control algorithms

In the following we describe the different robot control algorithms that we use to modify the motion behaviors of robots that, in turn, changes their interaction times.

---

**Algorithm 1:** Standard robot control algorithm for collective decision-making using the majority rule.

---

```

1 select initial opinion (uniform  $(A,B)$ );
2 while not(end of experiment) do
3   broadcast own opinion;
4   if  $Pr_d < Pr_\epsilon$  then
5     switch opinion spontaneously;
6   else
7     collect neighbors' opinions;
8     if # neighbors is even then
9       keep opinion;
10    else
11      adapt majority opinion;
12  diffuse with obstacle avoidance;

```

---

Robots move in a gas-like diffusion as they move away from areas of high robot density to spaces of low density. This motion mechanism serves as both, obstacle avoidance and exploration. In each simulation step, a robot collects vectors from its proximity sensors<sup>2</sup> consisting of relative distance and relative angle. A summed vector is used as indicator towards areas of low density (potential field method). If no other robots are detected, robots move in a straight line.

Following the standard control algorithm (Alg. 1), collective decision-making is implemented by the majority rule. We keep this decision algorithm unchanged for all experiments—since we focus exclusively on motion patterns and their interaction times. A robot collects the opinions of its neighborhood and switches to (or keeps) the opinion of the majority. We add a probabilistic element by spontaneous opinion switching with probability  $Pr_\epsilon = 0.1$  (otherwise majority rule with probability  $1 - Pr_\epsilon = 0.9$ ). Spontaneous switching helps to preserve the system's possibility to explore even when consensus is achieved. The decision-making process is continuous as it is active while the robots move around randomly, see Alg. 1.

We propose an algorithm to provoke a more mixed system by modifying the robot motion behavior. The idea is to increase the average number of neighbors and to reduce interaction times. Note that any opinion exchange requires a minimal time to transmit information. We assume this minimal time to be one discrete time step in our experiments (i.e., instantaneous opinion exchange). We call this variant the ‘well-mixed algorithm,’ see Alg. 2. A robot attempts to move away from its neighbors, once there can be no relevant information exchange anymore (i.e., homogeneous neighborhood). A robot computes a repulsive velocity vector using relative distances and angles to its neighbors. A robot is attracted by neighbors with opposing opinion. Hence, all robots try

<sup>2</sup>the simulated Footbot uses 24 proximity sensors

---

**Algorithm 2:** Well-mixed robot control algorithm for faster opinion sampling in collective decision-making.

---

```

1 select initial opinion (uniform  $(A,B)$ );
2 while not(end of experiment) do
3   broadcast own opinion;
4   if  $Pr_d < Pr_\epsilon$  then
5     switch opinion spontaneously;
6   else
7     collect neighbors' opinions;
8     if # neighbors is even then
9       keep opinion;
10    else
11      adapt majority opinion;
12  if  $N_{diff} > 1$  then
13    for  $i=1, i \leq N_{diff}$  do
14       $x = \text{sensor}(i).\text{value} * \cos(\text{sensor}(i).\text{angle});$ 
15       $y = \text{sensor}(i).\text{value} * \sin(\text{sensor}(i).\text{angle});$ 
16       $v.x = v.x + x;$ 
17       $v.y = v.y + y;$ 
18       $\alpha_{\text{attract}} = \text{atan2}(v.y, v.x);$ 
19      compute linear speed from  $\alpha_{\text{attract}};$ 
20    else if all neighbors with same opinion then
21      for  $i=1, i \leq N_{all}$  do
22         $x = \text{sensor}(i).\text{value} * \cos(\text{sensor}(i).\text{angle});$ 
23         $y = \text{sensor}(i).\text{value} * \sin(\text{sensor}(i).\text{angle});$ 
24         $v.x = v.x + x;$ 
25         $v.y = v.y + y;$ 
26         $\alpha_{\text{repulse}} = \text{atan2}(v.y, v.x);$ 
27      compute linear speed from  $\alpha_{\text{repulse}};$ 
28    else
29      diffuse with obstacle avoidance;

```

---

to maximize interaction between robots of different opinion. On the one hand, robots maximize their dissemination effect by approaching and convincing minorities of robots with opposing opinion. On the other hand, robots maximize chances to be convinced of the opposite by their exposure to robots of opposing opinion. The algorithm pushes robots from one neighborhood to another avoiding to waste time in homogeneous sub-populations. As mentioned above, once consensus is reached robots don't need to maintain a certain motion pattern for opinion mixture. In fact, that is a waste of energy. We propose Alg. 3 as the ‘energy-aware’ algorithm. Robots exchange opinions in their local neighborhood over time and use their collected sample to decide when they can stop their specific motion pattern.

Finally, we propose a robot control algorithm, that tries to contain contrarians in collective decision-making (see Alg. 4) called ‘impeding algorithm’. We have two sub-populations: (1) the standard individuals executing the majority rule (as

---

**Algorithm 3:** Robot control algorithm for adaptive interaction time based on individual awareness of coherence degree in collective decision-making.

---

```

1 while not(end of experiment) do
2   Apply decision making using Alg. 1;
3   for  $i=1, i \leq N_{neighbors}$  do
4     if  $opinion(i)=A$  then
5        $\#Neighbors\_A=\#Neighbors\_A+1$ ;
6     else if  $opinion(i)=B$  then
7        $\#Neighbors\_B=\#Neighbors\_B+1$ ;
8    $Neighbors\_Composition=|\#Neighbors\_A-$ 
    $\#Neighbors\_B|/size\_local\_neighborhood$ ;
9   if  $simulation\_timer \% \delta\_t$  then
10    compute mean  $\mu_{neighborhood}$  of
     $Neighbors\_Composition$ ;
11    if  $\mu_{neighborhood} = 1$  then
12      start slowing down by  $\delta\_s$  at each time step;
13      reset  $Neighbors\_Composition$ ;
14    diffuse with obstacle avoidance at the updated speed;

```

---

described in Alg. 1), (2) the contrarians that always adopt the minority opinion of the neighborhood. Our goal is to limit the effect of contrarians by containing them. Following the impeding algorithm, we modify the motion of standard robots who are neighbors of contrarians. Neighbors of a contrarian (who may already be affected by its opinion) try to increase their interaction time with it to prevent contact with other robots. When standard robots find a contrarian, they encircle it in rough analogy to immune systems (see footnote 3).

## Results

We validate our algorithms by running physics-based simulations using the ARGoS simulator (Pinciroli et al., 2012). We set an arena of  $6 \times 8 m^2$  and test swarm sizes  $N \in \{50, 100, 200, 300, 400, 500\}$  in a symmetry-breaking collective decision task. Each experiment run is repeated 30 times independently with 2000 time steps each. Tab. 1 gives the parameter values of both the robot and control algorithms.

### A - interaction times as function of swarm density

We study the relation between agent-to-agent interaction times and swarm densities (robots per area). Knowledge of this relation can be used to modulate interaction times offline by setting appropriate swarm densities. Due to limited space, increased swarm densities may cause cascades of collision avoidance actions, where robots take longer to leave their neighborhoods. In Fig. 1a, we show the average interaction times over swarm size. We keep the area constant,

---

**Algorithm 4:** Impeding robot control algorithm to contain contrarians in collective decision-making.

---

```

1 while not(end of experiment) do
2   broadcast own opinion;
3   if  $Pr_d < Pr_\epsilon$  then
4     switch opinion spontaneously;
5   else
6     collect neighbors' opinions;
7     if  $\#neighbors$  is even then
8       keep opinion;
9     else if  $(\#neighbors\ odd) \ \& \ (not\ contrarian)$ 
10      then
11        adapt majority opinion;
12      else if  $(\#neighbors\ odd) \ \& \ (contrarian)$  then
13        adapt minority opinion;
14  if  $\#contrarian\ neighbors > 0$  then
15    for  $i=1, i \leq N_{contrarian}$  do
16       $x = sensor(i).value * \cos(sensor(i).angle)$ ;
17       $y = sensor(i).value * \sin(sensor(i).angle)$ ;
18       $v.x = v.x + x$ ;
19       $v.y = v.y + y$ ;
20     $\alpha_{attract\_to\_contrarian} = \text{atan2}(v.y, v.x)$ ;
21    compute linear speed from  $\alpha_{attract\_to\_contrarian}$ ;
22  else
23    diffuse with obstacle avoidance;

```

---

Table 1: Parameters used in the simulation.

Parameter	Value
<i>Robot parameters</i>	
Type	Footbot
Proximity sensor range $r_{prox}$	0.1 m
Range-and-bearing sensor range $r_{rab}$	1.0 m
Maximum moving speed	$5 \frac{m}{s}$
<i>Algorithms parameters</i>	
<i>Alg. 1</i>	
$Pr_\epsilon$	0.1
<i>Alg. 3</i>	
$\delta\_t$	30 s
$\delta\_s$	$0.01 \frac{m}{s}$

hence, increased swarm size means increased swarm density. Average interaction times and their standard deviation increase super-linearly with increasing swarm size. In Fig. 1b, we show average interaction times as a function of average distance between robots. The values are averaged over all robots and 30 independent runs each. In full correspondence to Fig. 1a, we observe a decrease of interaction times with increased robot-to-robot distance.

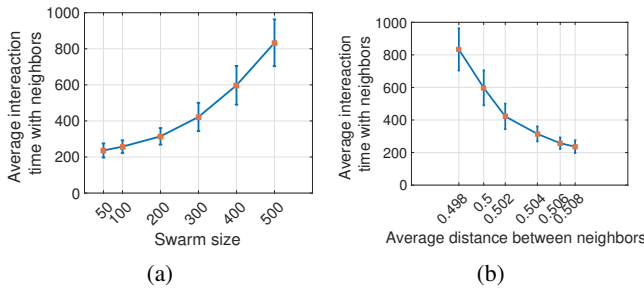


Figure 1: The average interaction times as a function of (a) the population size/density. (b) The average distance between robots for different densities. Error bars give the standard deviation over independent 30 runs.

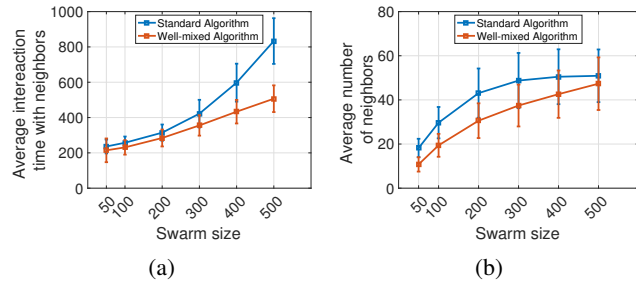


Figure 2: Comparison well-mixed to standard algorithm. The average of (a) interaction times and (b) neighborhood size is computed as a function of the swarm size/density. Error bars give the standard deviation over 30 runs.

## B - boosting spread of desired information

In this set of experiments, robots use the majority rule as decision mechanism (Alg. 1) with a probability of  $Pr_\epsilon = 0.1$  for spontaneous option switching. This implements an exploration behavior to keep the swarm adaptive. First, we exploit only the robots' (gas-like) diffusion mechanism that we used above to generate the interaction time (Fig. 1). Second, we validate our well-mixed algorithm.

In Fig. 2a we compare the interaction times for a swarm controlled by the well-mixed algorithm to a swarm controlled by the standard algorithm. Especially for large swarms (e.g.,  $N = 400$  and  $N = 500$ ), we notice a significant reduction of interaction times using the well-mixed algorithm. In Fig. 2b we compare the resulting average neighborhood sizes. There is a trend to reduced neighborhood sizes using the well-mixed algorithm, specifically for medium swarm sizes (e.g.,  $N = 200$  and  $N = 300$ ). For large neighborhood sizes (e.g.,  $N = 400$  and  $N = 500$ ) both algorithms approach saturated neighborhood sizes of  $\approx 48$  robots. This is the limit due to densely packed clusters of robots.

In Fig. 3a, we compare the well-mixed to the standard algorithm in terms of time to convergence. We define time to convergence using a threshold as the first passage time to (absorbing) states of more than 90% majority. In our comparison, we notice that with increasing swarm size, robots spend

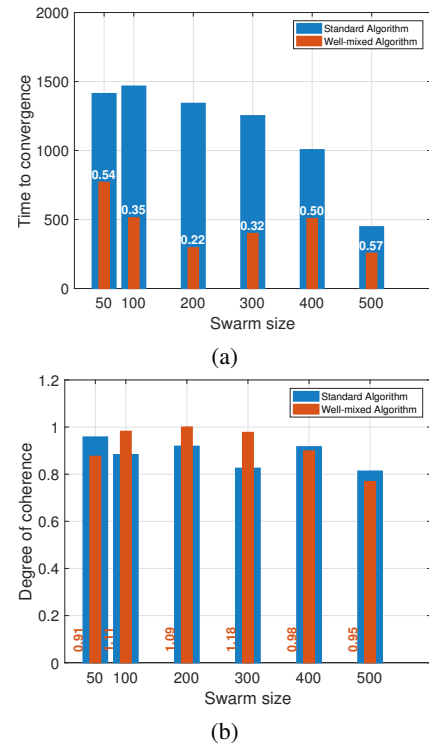


Figure 3: (a) The time the swarm takes to converge to an equilibrium, (b) the degree of coherence reached by the swarm. Both computed over the different population densities, and under both algorithms (i) standard, and (ii) well-mixed.

more time to avoid obstacles and they cover less distance. Due to the high density, swarm-wide connectivity is highly probable resulting in large connected components. The time to propagate opinions is reduced compared to sparse densities when robots need to move around in 'empty space' to connect to new neighborhoods. The result is a decreasing time to convergence with increasing swarm size. Our proposed well-mixed algorithm reduces the time to convergence significantly compared to the standard algorithm. The improvement (reduction of time to converge) reaches a maximum of 78% in the case of medium density ( $N = 200$  robots), and a minimum of 43% in the case of dense swarms ( $N = 500$  robots). Best improvements are achieved for medium swarm sizes ( $100 \leq N \leq 300$ ). With increasing density (e.g.,  $N = 500$ ) robots cannot move anymore and hence no further improvement is possible.

We use the degree of coherence as a second method to measure performance. In Fig. 3b, we compare the degree of coherence for both algorithms (i.e., the standard and the well-mixed). For high densities ( $N \geq 400$ ) coherence doesn't differ between the two algorithms because the swarms are fully connected. Similarly for sparse swarms ( $N = 50$ ), robots mix well following the standard algorithm due to sparsely populated space. For low density ( $N = 50$ ) the well-mixed algorithm is even detrimental to coherence but it converges

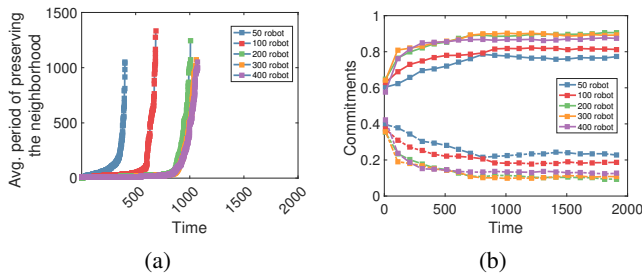


Figure 4: Results for the energy-efficient Alg. 3 for different swarm sizes. (a) average lifetime of a neighborhood (sub-population of the swarm), (b) commitment to the two opinions over time.

faster. This is because, in sparse swarms the well-mixed algorithm maximizes robot diffusion, that decreases robot interactions compared to the standard algorithm. The well-mixed algorithm enables the degree of coherence in medium densities ( $100 \leq N \leq 300$ ) to reach one asymptotically—i.e., full consensus. The improvement is between 9% and 18% compared to the standard approach.

Next, we study situations towards the end of runs when high coherence is achieved and it gets irrelevant to maintain a good mixture of opinions. When robots get aware of a high coherence by local measurements, they reduce their motion and hence increase their interaction time with their local neighborhood. We use Alg. 3 that implements this energy-efficient mechanism for collective decision-making. In Fig. 4a, we show the average time of how long a certain neighborhood configuration (a sub-population of the swarm) is preserved during the simulation. At the beginning of each simulation the neighborhood changes often (i.e., short life-times of neighborhood configurations and short interaction times). These times increase gradually over time as the system achieves higher degrees of coherence. Robots slow down their nominal velocity by  $\delta_s = 0.01$  at each time step when observing full consensus in their neighborhood for a time window of 300 time steps. As a result neighborhood configurations are preserved for longer times until a large majority of the swarm stops moving and builds stable neighborhoods, see Fig. 4a. The degree of coherence reached using Alg. 3 is high for all swarm sizes as seen in Fig. 4b. We also notice that neighborhoods freeze across almost all sizes of populations after the swarm has reached a high degree of coherence, except for swarm size  $N = 50$ . For this sparse swarm we would require larger time windows (i.e.,  $> 300$  time steps) to sample enough neighbors and have a more accurate decision about the degree of coherence reached at the system level.

### C - impeding spread of undesired information

Next, we study collective decision-making with sub-populations of contrarian robots. As mentioned above, contrarians observe the current majority in their neighborhood and then switch to the opinion of the minority. In general,

in collective systems we assume that sharing information is necessary and useful or at least not harmful. This implies that maximizing the number of encountered peers is a key objective. Minimizing the agent-to-agent interaction time or limiting it to the duration needed to exchange the information becomes the desired design feature. Nevertheless, in this setup we consider contrarians, whose influence may harm the system by preventing it from achieving a consensus. For such setup, spreading the contrarians’ opinions need to be limited. In a first set of experiments, we study symmetry-breaking when a sub-population of contrarians with percentages of 5%, 10%, and 30% is introduced.<sup>3</sup> All robots use the gas-like diffusion motion to wander in the arena. The majority of robots (95%, 90%, and 70%) runs Algo. 1 based on the majority rule while contrarians follow their ‘minority rule.’ Fig. 5a-e shows, respectively, how the swarm loses its capability of making a decision caused by the contrarian sub-population for percentages of 10% and 30%.

In a second set of experiments, we tune the interaction times of the contrarians’ neighbors in order to contain those and hence limit their influence to the system, see Alg. 4. We ignore aspects of how to detect and identify contrarians as this is out of scope of our study. Methods of fault detection could be applied (Lau et al., 2011; Tarapore et al., 2015). Here, we assume that robots detect contrarians immediately. Following our impeding algorithm, when a robot encounters a contrarian, it slows down to increase its interaction time with that contrarian. For a neighborhood with more than one contrarian, the robot computes a summed vector as an indicator towards containing all contrarians in the neighborhood. As soon as enough neighbours are available, they build up a cluster around that contrarian and isolate it from the rest of the swarm. This behavior triggered in the neighborhood of the contrarian restricts its influence by keeping it in the same neighborhood for long. In Fig. 5a, we show a swarm with a small sub-population of contrarians (5%) that have no considerable impact on the capability of the collective decision-making system to break the symmetry. This ability is lost when increasing the size of the contrarians sub-population to 10% and 30% as seen in Figs. 5b and d. Figs. 5c and e show the system’s improved capability to make decisions when applying the containing algorithm. Clearly the swarm performs better for a contrarian population of 10%. The improvement is minimal, however, for a contrarian population of 30%. This is because a large portion of the swarm is busy forming clusters around contrarians and too few robots are left for the actual decision-making task.

Fig. 6 shows the distributions of interaction times for the standard and impeding algorithm. For high density ( $N = 400$ ), the interaction times are similar but for all smaller tested swarm sizes ( $N < 400$ ) the impeding algorithm increases interaction times significantly. We notice a wide interval of

<sup>3</sup>video online: <https://youtu.be/zjklEseERak>

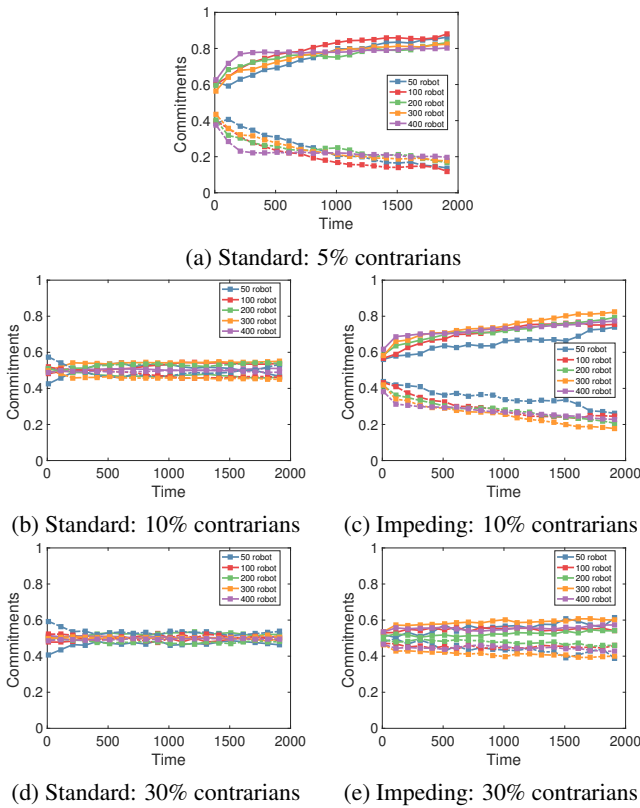


Figure 5: Commitment to the two options over time in the collective decision-making system using majority-rule (a, b, d) with the standard algorithm, (c, e) with the impeding algorithm.

interaction times using the impeding algorithm. This is a clear indicator of the emergence of sub-communities with long interaction times due to the impeding behavior.

Additionally, we compare the standard with the impeding algorithm using the histograms of interaction times in Fig. 7, for different swarm sizes. The histograms clearly indicate a bimodal distribution for the impeding algorithm (red) corresponding to the two sub-populations of contrarian containers and freely moving standard robots. For lower swarm densities ( $N \leq 200$ ), the left peak of the bimodal distribution, that represents the median of the interaction times among the standard robots, overlaps with the peak of the standard algorithm (blue). The right peak is generated by containing and contrarian robots and indicates higher interaction times. Interestingly, for increasing swarm density, the left peak moves to the left relative to the blue peak indicating shorter interaction times than for the standard algorithm (while the median of the complete bimodal distribution of 595.75 is similar to the median of the standard algorithm, 581.97). This is the impeding algorithm’s effect of increasing free space for motion of standard robots. Containing robots densely cluster with contrarians consuming only a small area. This allows the standard robots (left peak) to move quickly and to experience shorter interaction times.

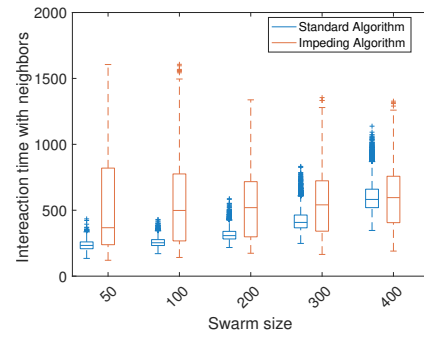


Figure 6: Robot interaction times over swarm size for the standard and the impeding algorithm. Notice the widely spread interaction times for the impeding algorithm.

## Conclusions

With focus on interaction times, we have shown for the example of collective decision-making that time to convergence can be improved by carefully designing the motion patterns of robots. The mixture of robots with different opinions can be increased by minimizing interaction times. Following the city metaphor, we create a hyperactive metropolis. In the case of malicious robots (here contrarians), we do not require sophisticated methods of forming and administrating trust networks. Instead, we have shown that significantly increased interaction times with these robots—by exploiting a simple containing strategy designed based on robot motion patterns—can isolate them and reduce their undesired influence. Following the city metaphor, we force contrarians into unhurried spots (e.g., villages) reducing their ability to spread information. With the proposed energy-aware approach, we then regulate interaction times online depending on the level of coherence achieved by the society. Following the city metaphor, this may correspond to a weekly rest day when city life is reduced to a minimum, with the only difference that here the robots reach global awareness of when to schedule that. In future work we plan to test these algorithms on real robots.

## References

- Christakis, N. A. and Fowler, J. H. (2007). The spread of obesity in a large social network over 32 years. *New England journal of medicine*, 357(4):370–379.
- Gambaro, J. P. and Crokidakis, N. (2017). The influence of contrarians in the dynamics of opinion formation. *Physica A: Statistical Mechanics and its Applications*, 486:465–472.
- Hamann, H. (2018a). Opinion dynamics with mobile agents: Contrarian effects by spatial correlations. *Frontiers in Robotics and AI*, 5:63.
- Hamann, H. (2018b). *Swarm Robotics: A Formal Approach*. Springer.
- Hamann, H., Schmickl, T., Wörn, H., and Crailsheim, K. (2012). Analysis of emergent symmetry breaking in collective decision making. *Neural Computing & Applications*, 21(2):207–218.

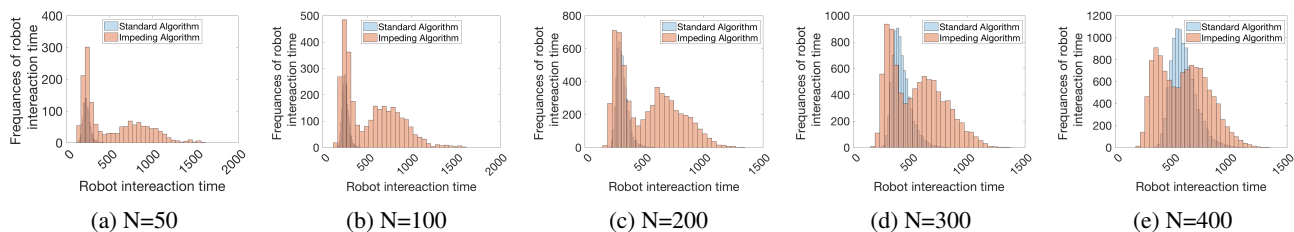


Figure 7: Distribution of interaction times over different swarm sizes, for the standard algorithm and the impeding algorithm, for the latter: bimodal distributions associated with the two sub-populations of contrarian containers and standard robots.

- Hamann, H., Valentini, G., Khaluf, Y., and Dorigo, M. (2014). Derivation of a micro-macro link for collective decision-making systems. In *International Conference on Parallel Problem Solving from Nature*, pages 181–190. Springer.
- Heesterbeek, H., Anderson, R. M., Andreasen, V., Bansal, S., De Angelis, D., Dye, C., Eames, K. T., Edmunds, W. J., Frost, S. D., Funk, S., et al. (2015). Modeling infectious disease dynamics in the complex landscape of global health. *Science*, 347(6227).
- Hess, A., Hummel, K. A., Gansterer, W. N., and Haring, G. (2016). Data-driven human mobility modeling: a survey and engineering guidance for mobile networking. *ACM Computing Surveys (CSUR)*, 48(3):38.
- Keeling, M. J. and Rohani, P. (2011). *Modeling infectious diseases in humans and animals*. Princeton University Press.
- Khalil, N. and Toral, R. (2019). The noisy voter model under the influence of contrarians. *Physica A: Statistical Mechanics and its Applications*, 515:81–92.
- Khaluf, Y. (2017). Can the mechanisms of self-organization help our cities in following the scaling laws of biology? In *Proceedings on the International Conference on Artificial Intelligence (ICAI)*, pages 41–44. The Steering Committee of The World Congress in Computer Science, Computer .
- Khaluf, Y., Pinciroli, C., Valentini, G., and Hamann, H. (2017). The impact of agent density on scalability in collective systems: noise-induced versus majority-based bistability. *Swarm Intelligence*, 11(2):155–179.
- Khaluf, Y., Rausch, I., and Simoens, P. (2018). The impact of interaction models on the coherence of collective decision-making: A case study with simulated locusts. In *International Conference on Swarm Intelligence*, pages 252–263. Springer.
- Krings, G., Karsai, M., Bernhardsson, S., Blondel, V. D., and Saramäki, J. (2012). Effects of time window size and placement on the structure of an aggregated communication network. *EPJ Data Science*, 1(1):4.
- Lau, H., Bate, I., Cairns, P., and Timmis, J. (2011). Adaptive data-driven error detection in swarm robotics with statistical classifiers. *Robotics and Autonomous Systems*, 59(12):1021–1035.
- Leskovec, J., Adamic, L. A., and Huberman, B. A. (2007). The dynamics of viral marketing. *ACM Tran. on the Web*, 1(1):5.
- Leskovec, J., Backstrom, L., and Kleinberg, J. (2009). Meme-tracking and the dynamics of the news cycle. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 497–506. ACM.
- Meier, R. L. (1962). *A Communications Theory of Urban Growth*. MIT Press.
- Orellana-Rodriguez, C. and Keane, M. T. (2018). Attention to news and its dissemination on twitter: A survey. *Computer Science Review*, 29:74–94.
- Pinciroli, C., Trianni, V., O’Grady, R., Pini, G., Brutschy, A., Brambilla, M., Mathews, N., Ferrante, E., Caro, G., Ducatelle, F., Birattari, M., Gambardella, L. M., and Dorigo, M. (2012). ARGoS: a modular, parallel, multi-engine simulator for multi-robot systems. *Swarm Intelligence*, 6(4):271–295.
- Polletta, F. and Jasper, J. M. (2001). Collective identity and social movements. *Annual review of Sociology*, 27(1):283–305.
- Pratt, S. C., Mallon, E. B., Sumpter, D. J., and Franks, N. R. (2002). Quorum sensing, recruitment, and collective decision-making during colony emigration by the ant *leptothorax albipennis*. *Behavioral Ecology and Sociobiology*, 52(2):117–127.
- Rausch, I., Khaluf, Y., and Simoens, P. (2018). Applying scale-invariant dynamics to improve consensus achievement of agents in motion. In *Int. Symp. on Distributed Computing and Artificial Intelligence*, pages 344–348. Springer.
- Stark, H.-U., Tessone, C. J., and Schweitzer, F. (2008). Decelerating microdynamics can accelerate macrodynamics in the voter model. *Physical review letters*, 101(1):018701.
- Stroeymeyt, N., Grasse, A. V., Crespi, A., Mersch, D. P., Cremer, S., and Keller, L. (2018). Social network plasticity decreases disease transmission in a eusocial insect. *Science*, 362(6417):941–945.
- Tarapore, D., Lima, P. U., Carneiro, J., and Christensen, A. L. (2015). To err is robotic, to tolerate immunological: fault detection in multirobot systems. *Bioinspiration & biomimetics*, 10(1):016014.
- Valentini, G., Ferrante, E., Hamann, H., and Dorigo, M. (2016). Collective decision with 100 Kilobots: Speed versus accuracy in binary discrimination problems. *Autonomous Agents and Multi-Agent Systems*, 30(3):553–580.
- Yang, Y., Chawla, N. V., Basu, P., Prabhala, B., and La Porta, T. (2013). Link prediction in human mobility networks. In *2013 IEEE/ACM Int. Conf. on Advances in Social Networks Analysis and Mining*, pages 380–387. IEEE.
- Zhong, X., Zhao, J., Yang, L.-X., Yang, X., Wu, Y., and Tang, Y. Y. (2018). A dynamic discount pricing strategy for viral marketing. *PloS one*, 13(12):e0208738.