

# On the Convergence and Clustering Dynamics of the Hegselmann-Krause System

**Sophie Decoppet**  
Department of Computer Science  
Stanford University  
sophied@stanford.edu

**Peter Hansel**  
Department of Computer Science  
Stanford University  
pwhansel@stanford.edu

**Benjamin Wittenbrink**  
Department of Computer Science  
Stanford University  
witten@stanford.edu

## 1 Introduction

The phenomenon of self-organizing collective behavior – that is, behavior that arises absent any global coordination, rather through the aggregation of locally optimal decision-making – has implications for many different applications in a variety of fields, from magnetization in classical physics to robot swarms in robotics to models of pricing in economics [1]. Opinion dynamics is often formulated as one of these processes, and the Hegselmann-Krause (HK) is a common model for this task [2]. This paper will explore the one-dimensional case and analyze both the convergence and clustering dynamics of the model. We take a simulation approach to both topics and prove a result providing an upper bound for the density of clusters. Finally, we discuss a modification to the model, altering the underlying connectivity of the agents.

## 2 Hegselmann-Krause Model

We proceed by formalizing the definition of the Hegselmann-Krause (HK) [2] system in dimension  $d$ . Let there be  $n$  agents. For every time step  $t \in \mathbb{Z}^{\geq 0}$ , the position of an agent  $i$  is given according to  $x_i(t) \in \mathbb{R}^d$ . Then  $x(t) = [x_1(t), x_2(t), \dots, x_n(t)]$  is taken to be the state of the system at time  $t$ .

We define the neighborhood of an agent  $i$  at time  $t$  as

$$\mathcal{N}_i(t) = \{j \in [n] : d(x_i(t), x_j(t)) \leq R\} \quad (1)$$

for a vision range  $R$  and a distance measure  $d$ . As is common practice in the literature, we let  $R = 1$  and  $d(a, b) = \|a - b\|$ , that is the Euclidean norm for the rest of the paper. Hence, the neighborhood of agent  $i$  at time  $t$  is the collection of agents within a distance of 1. Every  $j \in \mathcal{N}_i(t)$  is referred to as a neighbor and this relationship is symmetric,  $i \in \mathcal{N}_j(t)$  if and only if  $j \in \mathcal{N}_i(t)$ . Note that  $i$  is always a neighbor to itself. We refer to an agent  $i$  with  $|\mathcal{N}_i(t)| = 1$  as isolated at time  $t$ .

At  $t = 0$ , the position of every agent is fixed. Note for convenience we will often denote this initial state as  $x_0$ . The positioning of the agents is then subsequently synchronously updated at each step in discrete time, according to the following rule

$$x_i(t+1) = \frac{1}{|\mathcal{N}_i(t)|} \sum_{j:j \in \mathcal{N}_i(t)} x_j(t). \quad (2)$$

In words, the position of an agent  $i$  at time  $t+1$  updated to be the average position of all other agents in their neighborhood.

For the rest of this paper, we consider the one-dimensional Hegselmann-Krause system – that is,  $d = 1$ . Then each agent has a scalar position at each time step and the distance between agents is simply the absolute difference along the real number line.

Below we outline two important properties of the one dimensional system.

*Proposition 1* ([3]). The system preserves the order of positions. That is, for any  $i, j \in [n]$ , if  $x_i(0) \leq x_j(0)$ , then  $x_i(t) \leq x_j(t)$  for all  $t > 0$ .

*Proposition 2* ([3]). If at time  $t$ , and for some  $i \in [n]$ ,  $|x_{i+1}(t) - x_i(t)| > 1$ , then for all  $t' > t$  also,  $|x_{i+1}(t') - x_i(t')| > 1$ . So the system can be decomposed into subsystems, one consisting of agents  $\{1, \dots, i\}$  and the other of agents  $\{i + 1, \dots, n\}$ , each evolving independently after time  $t$ .

An important implication of Proposition 2 is that, if at any time  $t$ , an agent's neighborhood only includes itself, it will never move again. We call such a point *frozen* at time  $t$ . Convergence occurs when all points are frozen.

### 3 Convergence

One common quantity of interest in the HK literature is the number of steps required for the system to converge; that is, the time  $t$  such that for all  $i \in [n]$  we have  $x_i(t + 1) = x_i(t)$ . Given an initial state  $x_0$ , we define  $x_\infty(x_0) = \lim_{t \rightarrow \infty} x(t)$  and refer to  $x_\infty(x_0)$  as the steady state corresponding to initial state  $x_0$  [4]. Formally, we may then define the termination time as follows.

**Definition 3.1** (Termination Time [4]). Given an initial state  $x_0$ , the *termination time*  $T(x_0)$  is the number of time steps required for the system to converge to the steady state corresponding to  $x_0$ . That is,

$$T(x_0) = \inf\{t \in \mathbb{N} : x(t) = x_\infty(x_0)\}. \quad (3)$$

Most work in the literature has been concerned with obtaining an upper bound on the termination time. This was first established to be  $O(n^5)$  in [5], though [3] improved this result to a tighter bound of  $O(n^3)$ . The lower bound has received less attention, in part because it is difficult to construct examples that take a long time to converge. Consequently, the lower bound was believed to be  $\Omega(n)$  [5]. However, subsequently [6] shows that a specific configuration freezes only after time  $\Omega(n^2)$ .

Convergence is currently understood to lie between  $n^2$  and  $n^3$ . That is, independent of the initial state,  $O(n^2) \leq T(x_0) \leq O(n^3)$  for any  $x_0$ .

**Theorem 3.2** ([3]). *The HK system converges within  $O(n^3)$  time steps.*

We provide a short proof of this result.

*Proof.* From Proposition 1, we have that there exists an ordering over the agents such that  $x_i(t) \leq x_{i+1}(t)$  for all  $t$  and  $i \in \{1, \dots, n - 1\}$ . Consider time step  $t$  and suppose that the system has not yet converged. Then there must exist at least one agent  $i$  whose neighborhood consists of an agent  $j$  where  $x_i(t) \neq x_j(t)$ . Consider the leftmost such agent, denoted as  $\ell(t)$ . Note that by this minimality condition, this agent must have a neighbor strictly to its right and none to its left.

Before we proceed, we introduce the following lemma (whose proof we omit for brevity, see [3]).

**Lemma 3.3.** *For every  $t \geq 0$ , by time  $t + 2$ , agent  $\ell(t)$  increases in weight, or gets frozen, or moves to the right by at least  $\frac{1}{2n^2}$ .*

Without loss of generality,  $x_n(0) - x_1(0) \leq n$  because otherwise Proposition 2 would allow us to decompose the system into independent subsystems. Thus,  $x_n(t) - x_1(t) \leq n$  for all  $t$ . Now consider applying the lemma above until there is no more such agent  $\ell$ . Observe that since there are  $n$  points,  $\ell(t)$  can only increase in weight at most  $n$  times. Additionally, since  $\ell$  has no left neighbors, it must be non-decreasing over  $t$ . In particular, we must have  $x_{\ell(t+2)}(t + 2) \geq x_{\ell(t)}(t + 2)$ , and thus the third case in the lemma may only occur a maximum of  $2n^3$  times. Putting this together, we have that the lemma can only be applied  $2(n + 2n^3)$  times, and thus the system must have converged for  $t > 2(n + 2n^3)$ . Thus,  $T \leq O(n^3)$ .  $\square$

In this section, we study the convergence of the HK system according to different initializations  $x_0$  using a simulation approach. In particular, we analyze two configurations: equal spacing and the "dumbbell".

### 3.1 Equal Spacing.

First, we consider equal spacing, where each agent is equally spaced along the real number with spacing  $d \in [0, 1]$ . We begin by considering the scenario where  $d = 1$ , and thus each agent is spaced by their vision range. We will denote this configuration as  $\mathcal{E}_n$ . For simplicity, we will begin the sequence at 1; thus, the  $i$ -th element of  $\mathcal{E}_n$  is equal to  $i$ .

This choice of configuration feels quite natural and intuitively one we might expect to have slow convergence. Indeed, it is straightforward to see that if such a spacing were infinitely long in both directions, i.e.  $\mathcal{E}_{\pm\infty} = (\dots, -1, 0, 1, \dots)$ , this would not terminate ever, as each agent is balanced between an agent on their left and on their right. However, because the sequence is not infinite, the endpoints have no agent more extreme than them, and thus are pulled toward the model. This leads to a domino effect and the model terminates.

This intuition leads us astray, as it can be shown that such a configuration converges in approximately  $5n/6$  iterations. This is visualized in Figure 1.

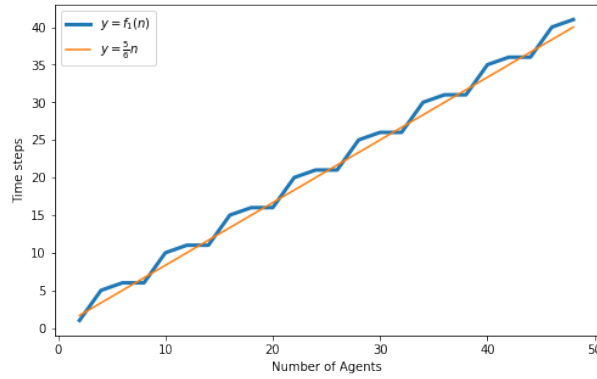


Figure 1: Convergence time of equal spaced agents as a function of number of agents

The graph demonstrates an interesting periodicity in the termination time. This stems from the fact that after every fifth step, a cluster of exactly three agents will disconnect and collapse in the subsequent step [7]. We illustrate this dynamic (on one end, as it is symmetric) in 2. We will analyze cluster formation more rigorously in a later section.

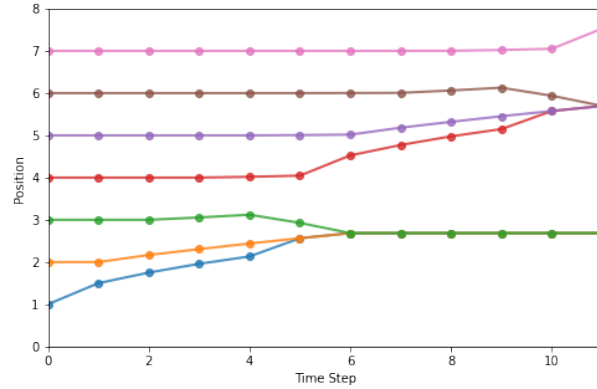


Figure 2: Example of cluster formation in equal spacing configuration with  $n = 90$

It has also been shown that there is a closed formula for the obtaining the termination time [7]:

$$T(\mathcal{E}_n) = 1 + 5 \lfloor \frac{n+2}{6} \rfloor + \frac{1}{3} (\sqrt{3} \sin(\frac{2\pi(n-1)}{3}) - \cos(\frac{\pi(n-1)}{3}) + 1). \quad (4)$$

One modification to this equal spacing configuration that preserves the spatial structure is to introduce weights at given points. Observe that we can equivalently formulate this as introducing additional

agents that occupy that position. For example, if we wanted to double the weight at an initial position in the equal spacing configuration, we could simply introduce another agent at that position. We observe that naively adding weights to points actually appears to accelerate the convergence. For example, consider the following procedure. Initialize  $n$  agents according  $\mathcal{E}_n$ . Then randomly choose a subset of size  $k$  agents to allocate  $n/k$  weight additional weight to. We analyze convergence as a function of both  $n$  and  $k$ . For fixed  $n$ , it is clear that increasing  $k$  substantially decreases the number of time steps until termination (see Figure 3). Moreover, even for low values of  $k$ , the number of iterations required is lower than in the unweighted case. The case scenario appears to be placing additional weight on the endpoints of  $\mathcal{E}_n$ .

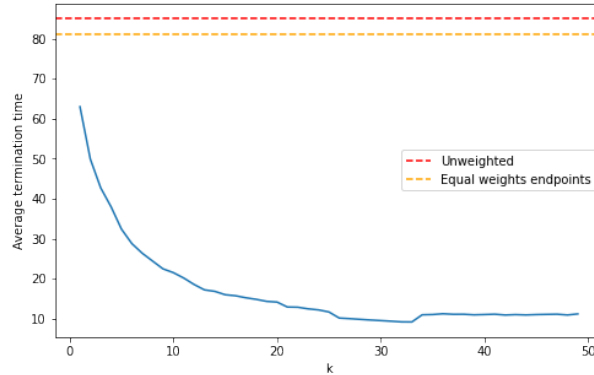


Figure 3: Average convergence time of equal spaced agents with weights placed randomly on  $k$  points

This makes intuitive sense, as in the unweighted version none of the “interior” points move by very much because they are always counterbalanced; only the endpoints are slowly being dragged to the middle ultimately cascading through the system. However, this parity is immediately broken when weights are imposed as points are drawn drastically in the direction of the biggest weight in their neighborhood.

So, how might we most adversarially add weights? The configuration discussed in the next section will provide an answer to this.

### 3.2 The Dumbbell Configuration.

Let  $n$  be a positive, even integer. The “dumbbell” configuration  $\mathcal{D}_n$  consists of  $3n + 1$  agents, whose positions are given as

$$x(i) = \begin{cases} -\frac{1}{n} & \text{if } 1 \leq i \leq n, \\ i - (n + 1) & \text{if } n + 1 \leq i \leq 2n + 1, \\ n + \frac{1}{n} & \text{if } 2n + 2 \leq i \leq 3n + 1. \end{cases} \quad (5)$$

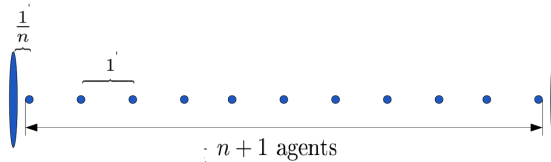


Figure 4: Depiction of the dumbbell configuration  $\mathcal{D}_n$  [3]

Thus, this configuration has a cluster of  $n$  agents at  $-1/n$ , then  $n + 1$  agents spaced one apart from 0 to  $n$ , and finally another cluster of  $n$  agents at  $n + 1/n$ . This yields a dumbbell shape, as it can be thought of the equal spacing formulation from above (the “bar”) with two masses on either side (the “weights”). See Figure 4 for a visual representation.

To illustrate the convergence dynamics of this system, we plot the evolution of agent positions in an example  $\mathcal{D}_4$  (13 agents in total) in Figure 5. The top and bottom lines are thicker as they represent

a greater point mass (these are the “weights” of the dumbbell). Convergence is symmetric around the midpoint of the interval, in this case that is  $n/2$ . Descriptively, the weights are pulled toward the middle because there is no stabilizing force on the other side as these are the endpoints. Eventually the weights are pulled sufficiently close that they start drawing the second closest agent as well. Once this happens, the domino effect starts and the agents begin to freeze.

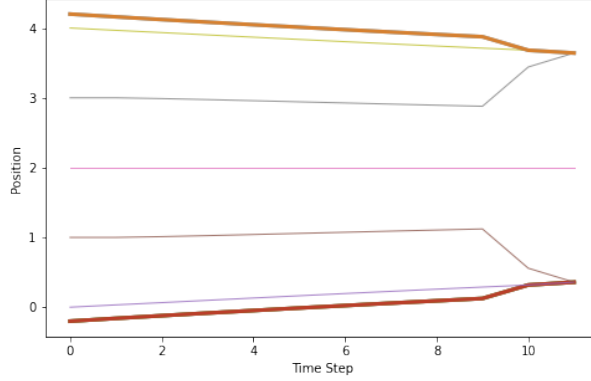


Figure 5: Example of convergence in dumbbell configuration for  $D_4$  (13 agents)

So, what is the time until convergence in this configuration? This question was answered by [6] and is presented in Proposition 3.

*Proposition 3* ([6]). The “dumbbell” configuration  $D_n$  has termination time  $T(D_n) = \Omega(n^2)$ .

To visualize this, we conduct a simulation study, calculating the number of time steps until convergence for every dumbbell  $D_n$  for  $n \in [2, \dots, 100]$ . We then plot these results (blue) in Figure 6 by the number of agents. Note, recall that  $D_n$  corresponds to  $m := 3n + 1$  agents. In addition, we plot the curve corresponding to  $0.03m^2$  (orange). Remarkably, this appears to be an incredibly accurate approximation of  $T(D_n)$ .

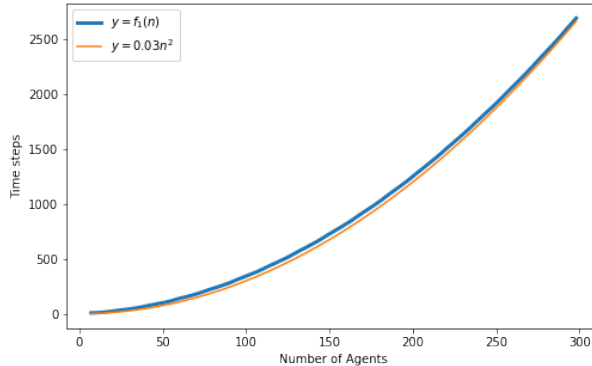


Figure 6: Convergence of dumbbell configuration

One open problem, as discussed in [6], is to compute the constant factor more accurately. The authors hypothesize that the termination time  $T(D_n) = (1 + o(1))\frac{n^2}{4}$ . We take an applied approach to this question to get an even more precise relationship. In particular, we run a linear regression according to the following specification

$$T(D_n) \sim \beta_0 + \beta_1 m + \beta_2 m^2 \quad (6)$$

on  $n \in [2, \dots, 50]$ , where  $\beta_2$  captures the coefficient on the quadratic term. We obtain the relationship  $T(D_n) = 0.0284m^2 + 0.575m + 0.6677$  and evaluate its predictive power in Figure 7. We note that the constant coefficient was substantially less significant than that of the higher-order terms. Specifically, we obtain  $\hat{T}(D_n)$  within in-sample (red points) and out-of-sample (blue points) on  $n \in [51, \dots, 200]$  (though for visual clarity only the first 50 points are plotted). Remarkably, this prediction is near perfect. Figure 8 visualizes the relative error  $\frac{\hat{T}(D_n) - T(D_n)}{T(D_n)}$  of the predictions for

the out-of-sample points. For larger  $n$ , it appears that the error converges to approximately 0.001 of the true termination time. Moreover, these values are positive, suggesting that we are overestimating the termination time.

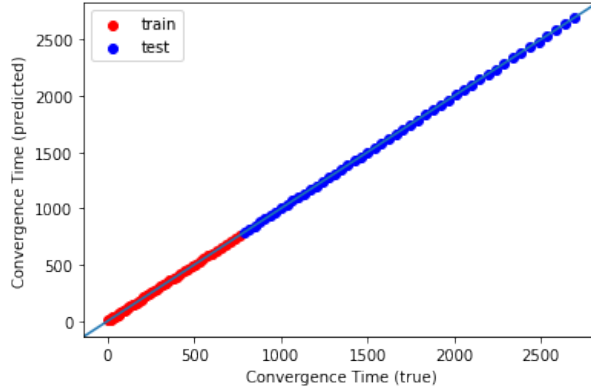


Figure 7: Evaluation of linear regression fit out-of-sample

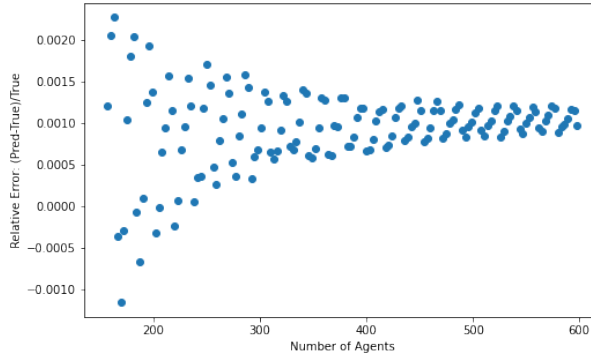


Figure 8: Relative error of linear regression out-of-sample

### 3.3 Width of the Interval.

In this section, we briefly explore the role that the width of the interval has in determining convergence. As previously discussed, it is significantly difficult to construct a configuration that requires greater than  $O(n)$  time steps. To our knowledge, the previous dumbbell configuration is the only known initialization to require  $\Omega(n^2)$ . Further complicating these dynamics, is that the width of the interval  $L$  appears to affect the time steps until termination non-linearly [8]. For example, clearly for  $L \leq 1$ , any configuration must converge in 1 step. Similarly, it can be shown that if  $L < 2$ , it must terminate in at most 2 steps. However, as can be seen below, as  $L$  increases, this relationship becomes increasingly difficult to parse.

To study this, we conduct the following experiment. Fix a value of  $L$  and  $n$ . Sample  $n$  points uniformly at random from the distribution  $[0, L]$ . Run the HK system until it terminates and note the time step  $t$ . We repeat this procedure 100 times for each pair  $(L, n)$  and take the average time required for termination. We plot the results in Figure 9.

As can be seen above, for most values of  $L$  the termination time appears to be sublinear in  $n$ . Meanwhile, when  $L = 5$ , the performance is dramatically worse and seemingly linear in  $n$ . Thus, we find that the convergence dynamics of the system also depend on the interval width. In the next section, we will explore how the interval width also determines the number of clusters.

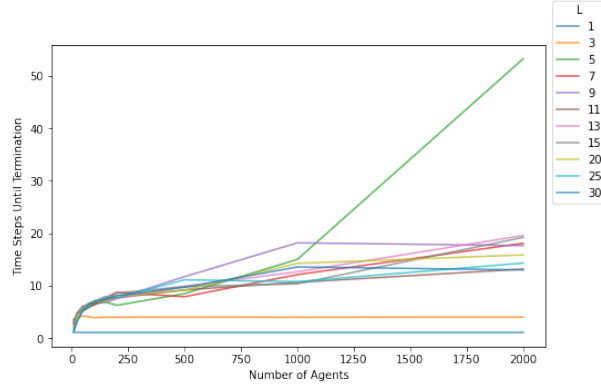


Figure 9: Average convergence time for uniform sampling along  $[0, L]$  with  $n$  agents

## 4 Clusters

In previous sections, we have mostly been concerned about how long it takes HK systems to converge. In this section we focus on what the state of convergence looks like, specifically the number of clusters in the final convergent state.

**Definition 4.1 (Clusters).** The *clusters* in the convergent state of an HK system is simply the values in  $x_\infty$  with duplicates removed. The size of this set is the number of clusters at convergence.

### 4.1 Clusters in our example configurations

Previously, we observed that with an equal spacing initialization, it seems that groups of three form into clusters in a 6-step process. We examine the process that makes this phenomenon occur consistently, by looking at the first 6 time steps for positions 1, 2, 3 in an equal space configuration (with sufficiently high  $n$ , for example in Figure 2. Call the agents agent  $a, b, c$  respectively

0. In the 0-th time step,  $a$  is at position 1,  $b$  at 2, and  $c$  at 3.
1.  $a$  moves to between 1.5 and 1.75,  $b$  and  $c$  don't move.
2.  $a$  moves to 1.75 and 2,  $b$  to between 2 and 2.2, and  $c$  doesn't move.
3.  $a$  moves to between 1.9 and 2,  $b$  to between 2.3 and 2.5, and  $c$  to between 3 and 3.2
4.  $a$  moves to between 2 and 2.5,  $b$  to between 2.4 and 2.6, and  $c$  to between 3 and  $a + 1$ . Then next agent (at position 4) moves up to something below 4.1.
5.  $a$  moves to between 2.5 and 3,  $b$  to between  $a$  and 3, and  $c$  to between  $b$  and 3. Then next agent (at position 4) moves up again, now more than  $c + 1$ . This starts this process at step 1 for the next 3 nodes as well, with an offset of 3 higher than positions and agents given here.
6.  $a, b, c$  converge to the same position between 2 and 3.

Because of the recursive nature of this process, this will repeat for every group of 3 agents. This is mirrored from the top as well, until this process meets in the middle. Thus, we see that the HK system converges with almost exactly  $L/3$  clusters (here,  $L = n - 1$ ).

Interestingly, we get an almost identical result for the dumbbell configuration as well. This is because the majority of the space is taken by uniformly-separated agents, just as the example we just covered. Intuitively, the extra agents at the end "slow down" convergence, but the clusters at the end look similar. We include in Figure 10 the graph of the number of clusters in each of the equal spacing and dumbbell configurations as a function of the initial interval  $L$ .

### 4.2 2R-Conjecture

Wedin [8] introduces what he calls the  $2R$ -conjecture (also see [9]). This involves counting clusters upon convergence of randomly-initialized graphs. Here,  $R$  is the vision range as we've described

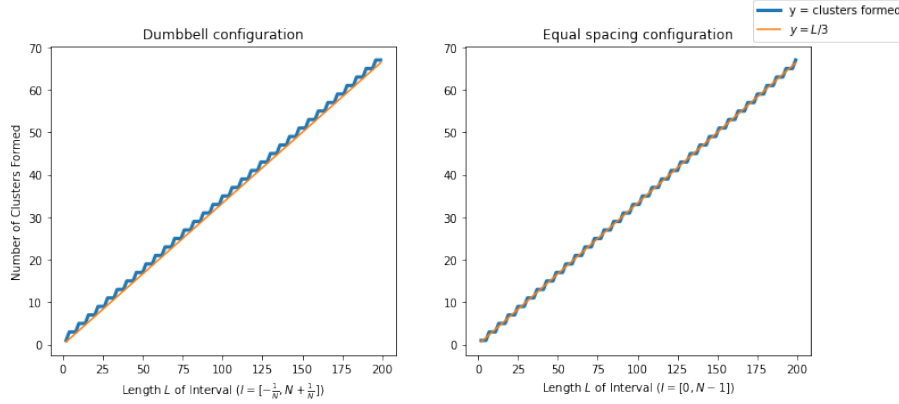


Figure 10: Clusters after convergence with dumbbell (left) and equal space (right) initializations, compared with  $L/3$

previously, and so we interpret it as 1. This conjecture states that the average distance between clusters upon uniform distributions tends to be more than 2. In his exploration, Wedin finds that the average distance between clusters with 5000 agents on the interval  $[0,100]$  is 2.4. Wedin laments that "this behavior is far from understood" and that perhaps "conjecture" is too strong. Nevertheless, it seems that the uniform random initialization yields on average more clusters than the uniform and dumbbell initializations.

This begs the question: what is the upper bound on the number of clusters that can be created over an initialization range  $L$ ?

### 4.3 Our Contribution: Dense Inputs and Dense Clusters

We first establish a very basic lower bound: we cannot have any clusters with distance 1 or less in the convergent state. We don't prove this rigorously, but it should be clearly intuitive: if we consider the "lowest" cluster that is within distance 1 or less of another cluster, that cluster will move toward the higher cluster. The result follows from the number of agents being finite. So what is the best we can do? Of course, we could initialize agents to be distance  $1 + \varepsilon$  apart to start (for any  $\varepsilon > 0$ ) and no other agents - this is immediately converged with distances arbitrarily close to 1. Thus, we focus on requiring there to be a high number of agents; specifically, agents must cover the interval arbitrarily densely. We prove the following result:

**Theorem 4.2.** *Let  $\delta > 0$  and  $\varepsilon > 0$ . For any  $L > 0$ , we can define an initialization that has points that fills the range  $[0, L]$  such that there is no continuous unfilled space of length more than  $\delta$  and upon convergence, there are at least  $\frac{L}{1+\varepsilon}$  clusters remaining.*

Our proof is constructive, so with this theorem we give a way to fill an interval  $[0, L]$  arbitrarily densely to converge with  $L$  clusters by setting  $\varepsilon = 1/L$ .

Our proof will look as follows: we construct enough agents to fulfill the requirement that there is no empty space greater than  $\delta$ . Then, we create large groups of points at intervals of  $1 + \varepsilon$ . At a high level, the first set of points is small enough that the space between the large groups doesn't change much, and they converge as clusters almost exactly as they started. However, if we set each of the large groups to have the same magnitude, most of the points between them settle in the middle, causing some of the groups to slowly converge. This phenomenon is illustrated in Figure 11. Thus, we need to be a little careful about how we initialize our large groups.

*Proof.* Let  $\delta, \varepsilon$  and  $L$  as given and construct the initialization as follows:

1. Initialize an agent at point  $n\delta$  for all integers  $0 \leq n \leq L/\delta$
2. Initialize  $\lceil \frac{801}{\delta^2 \varepsilon^2} \rceil$  agents at points  $2n(1 - \varepsilon)$  for  $0 \leq 2n \leq L/(1 + \varepsilon)$  where  $n$  is an integer



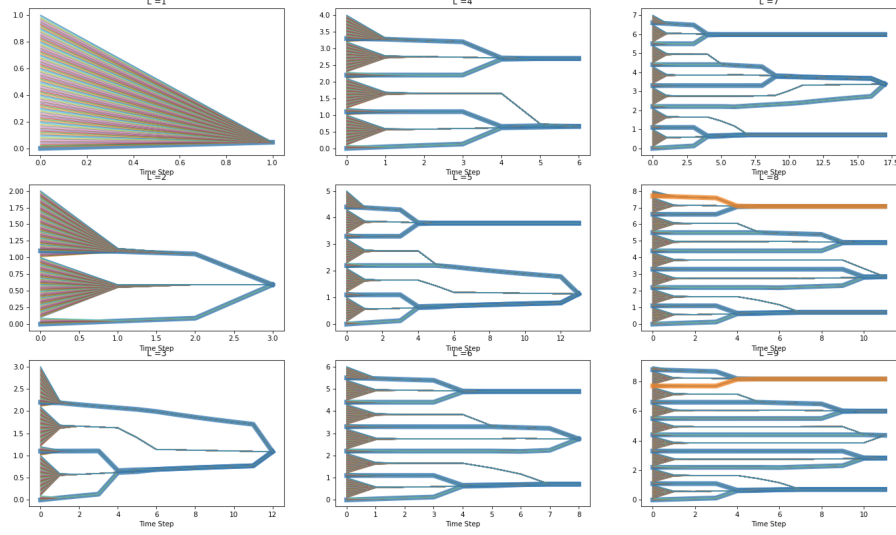


Figure 11: Convergence when initializing large groups of the same size for  $1 \leq L \leq 9$ . Notice that some of the large groups merge together after slowly moving toward an intermediate, smaller cluster.

3. Initialize  $\lceil \frac{4}{\delta \varepsilon} \rceil$  agents at points  $(2n + 1)(1 - \varepsilon)$  for  $0 \leq 2n + 1 \leq L/(1 + \varepsilon)$  where  $n$  is an integer

Steps 2 and 3 we will refer to as "large groups". Now, we look at the steps that lead to convergence:

1. In the first time step, any agent  $a$  initialized in step 1 will move to be very close to some large group, namely some group initialized in steps 2 or 3. There are two cases (assuming  $\varepsilon < 1$ ): either only one large group is within vision range of  $a$ , or two large groups are. In the former case, it will move within a range of less than  $\varepsilon/2$  of this group, since there are only at most  $2/\delta$  other points in its vision, and there are at least  $\lceil \frac{4}{\delta \varepsilon} \rceil$  agents in the large group. In the latter case, there will be one group of agents initialized in step 2 and one from step 3. Thus, there are  $\lceil \frac{801}{\delta^2 \varepsilon^2} \rceil$  agents in the largest group, plus an additional  $\lceil \frac{4}{\delta \varepsilon} \rceil$  agents in the other large group and  $2/\delta$  agents from step 1 in the vision of  $a$ . A proportion of at least  $\frac{100}{\delta \varepsilon}$  of these are in the group from step 2, so  $a$  will move within a range of  $\varepsilon/2$  from this group.

Now, the large groups are surrounded by points that are no more than  $\varepsilon/2$  away from the groups' original points. Since order must be preserved of the agents, the large groups cannot move much.

2. In the second time step, each large group will have points within  $\varepsilon/2$  of its original location, and nowhere else. Thus, each "pseudo-cluster" (it is not a cluster yet because we haven't reached convergence) has points no more than  $\varepsilon$  apart, and is more than 1 unit from any other pseudo-cluster. Thus, the pseudo-clusters merge, and we have convergence, because each now-cluster is distance more than 1 from any other cluster.

We see an illustration of this process in Figure 12. Indeed, large groups have two sizes, and barely move until convergence. From the fact that the large groups are surrounded by points no more than  $\varepsilon/2$  from their original location, they will not merge with other large groups as discussed in step 2. Thus, we have  $\frac{L}{1+\varepsilon}$  clusters, one for each large group created in steps 2 and 3. This completes our proof.  $\square$

This new result is interesting on its own; it does have the potential to be extended to include stronger assumptions - for example, it currently uses  $O(\frac{L}{\varepsilon^2 \delta^2})$  agents, which seems like it may be excessive. Additionally, this converges in 2 steps; it remains open if we can achieve  $L$  clusters after converging in more steps, for example  $O(n)$ .

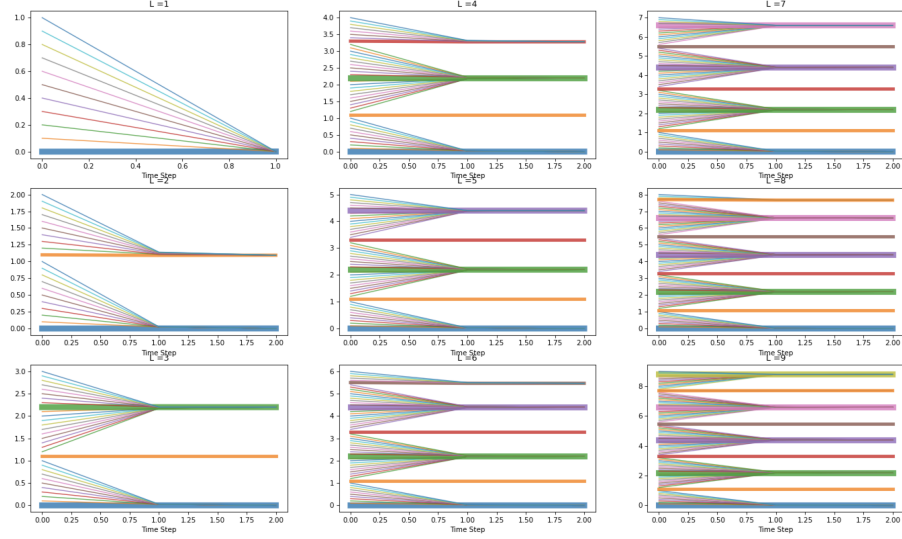


Figure 12: Convergence when initializing "large and larger" groups,  $1 \leq L \leq 9$ . Notice the alternating patterns of large and larger groups and that  $\#clusters = L$  upon convergence.

## 5 Graph Connectivity

### 5.1 Connectivity.

This section will discuss the notion of connectivity within the system and explore its impacts on convergence. Recall that in the standard HK model, agents update their positions based on the positions of those in their neighborhood. Returning to the model's original opinion dynamics framing, we can think about this as capturing the influence of an agent: that is, an agent is influenced by those in their neighborhood and not by those outside of it. However, there is always a potential or capacity for influence between any pair of agents.

However, what if we relax this assumption? That is, we incorporate the possibility that a given agent  $i$  will never be influenced by a certain agent  $i'$  even if  $i' \in \mathcal{N}_i(t)$  at a time step  $t$ . This relaxation makes a lot of sense intuitively as well as in application – there are certain people we choose to ignore or who we just do not have knowledge about and thus cannot be influenced by. Additionally, we will see that this has interesting effects on the convergence of the system.

To capture this, we introduce a physical connectivity graph  $G_{ph}$ , as in [4]. We define  $G_{ph}$  as an undirected graph on  $n$  vertices, such that each vertex corresponds to an agent. We extend an edge from a vertex  $i$  to a vertex  $j$  if and only if agent  $i$  has the potential to be influenced by agent  $j$ . In other words,  $(i, j) \in E(G_{ph})$  exists only when  $i$  has access to  $j$ . As  $G_{ph}$  is an undirected graph, this notion is symmetric.

We always assume that  $G_{ph}$  is connected and contains all self-loops – that is, an agent  $i$  always has access to their own position. Moreover,  $G_{ph}$  is assumed to be fixed for a given model and in particular is not dependent on the time, unless otherwise specified. Note, that applying this analysis to the original model, we would obtain  $G_{ph} = K_n$ , the complete graph on  $n$  vertices as every agent has access to every other agent's position.

With this representation of the connectivity of the system, we revisit the termination time definition from above.

**Definition 5.1** (Maximum Termination Time [4]). Given a physical connectivity graph  $G_{ph}$ , the *maximum termination time*  $T^*(G_{ph})$  is the supremum over the set of termination times corresponding to all possible initial states. That is,

$$T^*(G_{ph}) = \sup\{T(G_{ph}, x_0) : x_0 \in \mathbb{R}^n\}, \quad (7)$$

where we have extended Definition 3.1 to depend on both the initial state and the connectivity graph.

Furthermore, [4] shows that with this relaxation of connectivity, on average most HK systems no longer terminate.

*Proposition 4* ([4]). If  $G_{ph}$  is not a clique, the maximum termination time is infinite:  $T^*(G_{ph}) = \infty$ .

## 5.2 Redefining Convergence.

In the previous section, we have shown that limiting the connectivity of the system removes the guarantee of convergence to the steady state. However, this does not mean that the analysis of the convergence dynamics of such systems is fruitless. To proceed, we introduce the definition of  $\epsilon$ -convergence.

**Definition 5.2** ( $\epsilon$ -convergence [4]). For  $\epsilon > 0$ , given a physical connectivity graph  $G_{ph}$  and an initial state  $x_0 \in \mathbb{R}^n$ , a HK system reaches  $\epsilon$ -convergence at time  $T \in \mathbb{N}$  if its state is within  $\epsilon$  of the steady state corresponding to  $x_0$ , that is if

$$\|x(t) - x_\infty(x_0)\| < \epsilon, \quad (8)$$

for all  $t \geq T$ .

Observe that this is a weaker notion of convergence than previously used. Indeed, in this case, the state of the system must simply lie within an  $\epsilon$  neighborhood of the actual steady state. With this, we introduce the notion of  $\epsilon$ -convergence time.

**Definition 5.3** ( $\epsilon$ -Convergence Time [4]). For  $\epsilon > 0$ , given a physical connectivity graph  $G_{ph}$  and an initial state  $x_0 \in \mathbb{R}^n$ , the  $\epsilon$ -convergence time  $C_\epsilon(G_{ph}, x_0)$  is the number of time steps required for the system to achieve  $\epsilon$ -convergence, that is

$$C_\epsilon(G_{ph}, x_0) = \inf\{T \in \mathbb{N} : \|x(t) - x_\infty(x_0)\| < \epsilon, \forall t \geq T\}. \quad (9)$$

Then, just as with termination time, we can define the maximum  $\epsilon$ -convergence time.

**Definition 5.4** (Maximum  $\epsilon$ -Convergence Time [4]). For  $\epsilon > 0$ , given a physical connectivity graph  $G_{ph}$ , the *maximum  $\epsilon$ -convergence time*  $C_\epsilon^*(G_{ph})$  is the supremum over the set of  $\epsilon$ -convergence times corresponding to all possible initial states. That is,

$$C_\epsilon^*(G_{ph}) = \sup\{C_\epsilon(G_{ph}, x_0) : x_0 \in \mathbb{R}^n\}. \quad (10)$$

We now proceed with obtaining a lower bound on the  $\epsilon$ -convergence time as a function of the connectivity  $G_{ph}$ . To do so, we introduce the notion of conductance [10].

**Definition 5.5** (Conductance). Given an undirected graph  $G = (V, E)$  on  $n$  vertices, we define the conductance  $\phi(G)$  as

$$\phi(G) = \min_{S \subset V, S \neq \emptyset} \frac{|\partial(S)|}{\min(d(S), d(\bar{S}))}, \quad (11)$$

where  $\partial(S) = \{(i, j) \in E : i \in S, j \in \bar{S}\}$  and  $d(S)$  refers to the the sum of the degrees in the set.

Intuitively, this measures how “well-knit”  $G$  is. With this notion, we are able to obtain a lower bound on the maximum convergence time.

*Proposition 5* ([11]). Given an physical connectivity graph  $G_{ph}$  and  $\epsilon > 0$ , if  $G_{ph}$  is not isomorphic to  $K_n$ , the maximum convergence time  $C_\epsilon(G_{ph})$  satisfies

$$C_\epsilon(G_{ph}) > \frac{\log(\epsilon\sqrt{2})}{\log(1 - 2\phi(G_{ph}))}. \quad (12)$$

In words, this provides a lower bound on the convergence as a function of the conductance of the connectivity graph.

In this section, we study the convergence of the modified HK system according to different connectivity graphs using the equal spacing initialization.

### 5.3 Connectivity Simulations.

We first consider the dumbbell graph. This is defined similarly to the dumbbell initialization, in the sense that there are two large masses connected by a thin bar. In the case of the graph, we define it as being composed of two  $n/2$  cliques that are connected by a single edge. We note that  $\phi(D) = \frac{1}{n^2}$ , and so applying Proposition 5 yields a maximum convergence time of  $C_\epsilon^*(D) = \Omega(n^2)$  [4]. Notably, this is identical to the termination time for the dumbbell initialization in a fully connected graph, though these are separate concepts.

As a motivating example, consider a simple system with  $n = 20$  agents, each spaced 0.05 apart. As a result, for any agent  $i$ ,  $|x_i - x_j| < 1$  for all agents  $j$ , and thus  $N_i(0) = \{j : \forall j\}$ . Since every agent is a neighbor of every other agent, it follows that this system will terminate in a single iteration, as at  $t = 1$  an agent  $i$ 's position is given by

$$x_i(1) = \frac{1}{n} \sum_{k=1}^n x_k = x_j(1), \quad \forall j. \quad (13)$$

However, now consider this system but modify the connectivity to be given with a dumbbell graph. For simplicity, let the first ten agents  $[0, 0.05, \dots, 0.45]$  form the first clique and the remaining ten the second clique. Finally, extend an edge between the minimum agent in both subsets (i.e. the agent at 0 and at 0.5). Observe that after just one iteration, every member of a clique aside from the minimum member converges to the same value. However, because the minimum members are also connected to one another, they will be pulled closer toward the midpoint of the interval than the others. Note, that after the first iteration this problem essentially becomes a system with only four agents, two of which both have "weight" equal to  $(n - 2)/2$ . For simplicity, refer to the weighted agent below and above the midpoint as agents 1 and 4, respectively, and the non-weighted as agents 2 and 3. Because of the dumbbell connectivity, agents 1 and 4 are only influenced by 2 and 3, respectively, and thus are being very slowly dragged toward the midpoint. Note, there are no points to the left of agent 1 or to the right of agent 4, and hence thus after each time step they must move strictly toward the middle. In our simulation study, this simple system requires approximately 1,100 time steps to converge. The first 100 are plotted in Figure 13.

Alternatively, we can consider the star graph  $S_n$  (with self-loops), which is a bipartite graph of the form  $K_{1,n}$ . Thus, there is one vertex in the graph that is connected to all of the other vertices. Every other vertex is only connected to this center vertex and itself. This scenario is considerably faster than with dumbbell connectivity yet still much slower than full connectivity, as it converges in 32 time steps. [12] show that such a graph has a finite convergence time  $C_\epsilon^*(S_n) < \infty$ .

In fact, these example leads to a more general result, shown in [4].

*Proposition 6 ([4]).* If the initial configuration  $x(0)$  satisfies  $N_i(0) = \{j : \forall j\}$  for all  $i$ , then for any  $\epsilon > 0$ , the system achieves  $\epsilon$ -convergence in  $O(n^3 \log(n))$ .

Additionally, observe that it is not always the case that the dumbbell requires more time steps until convergence. If we consider the same exact system and connectivity graph, but with an initialization of agents spaced 1 apart, it is clear that the system would converge faster than the fully connected graph. This is because the two agents connecting the two cliques are never within range of another, and hence the two middle agents begin to diverge, as can be seen in 14.

In general, one counter-intuitive result that we encountered when designing these connectivity experiments is that increasing the connectivity of a graph does not ensure faster convergence. Indeed, [12] show that introducing edges into the graph has the possibility of moving  $C_\epsilon^*(G_{ph})$  from a finite value to  $\infty$ . Further complicating our understanding of the connectivity, is that convergence is defined over the set of all possible configurations, of which there are, of course, infinite. Hence, poor performance on one such configuration might lead to infinite convergence, even while performance in the average case is strong. This is a similarly predicament to the standard HK problem, where most configurations terminate in  $O(n)$ , yet careful constructions of the initial configuration like the dumbbell lead to the lower bound of  $\Omega(n^2)$ .

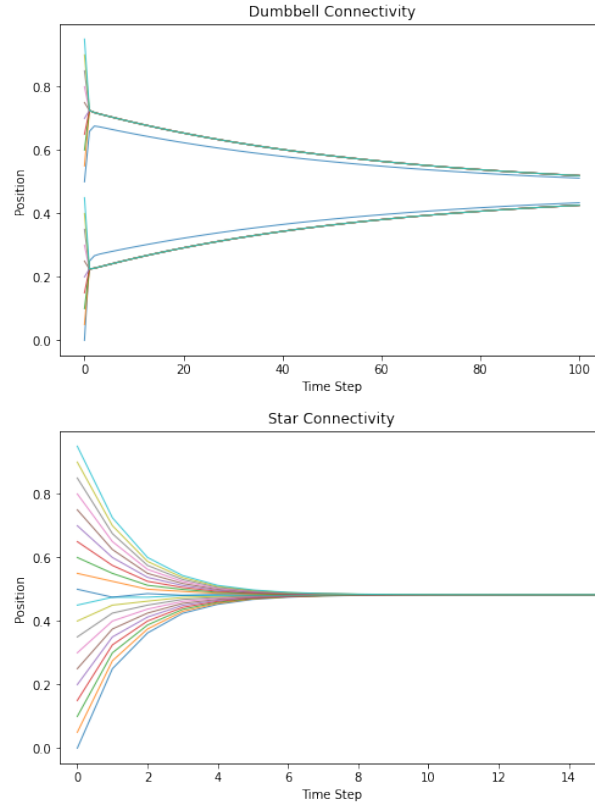


Figure 13: Example of convergence in equal spacing configuration with dumbbell (top) and star (bottom) connectivity,  $n = 20$

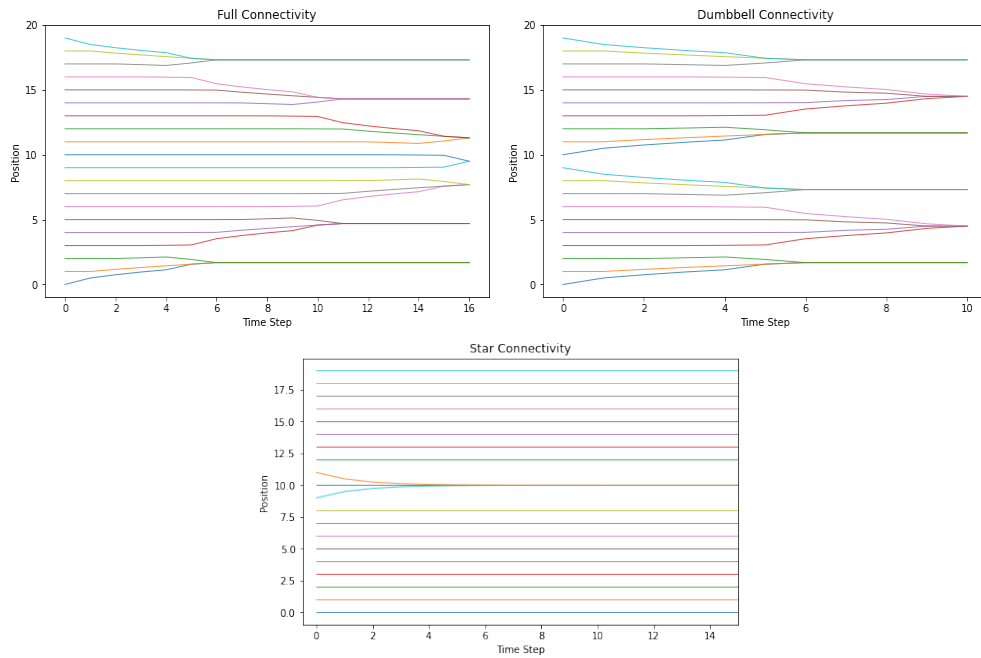


Figure 14: Example of convergence in equally spaced system ( $d = 1$ ), by connectivity  $n = 20$

## 6 Conclusion

In this paper, we have analyzed the convergence and clustering properties of the one-dimensional Hegselmann-Krause system. We run simulation experiments to analyze these dynamics and provide a proof of small clustering result.

Several directly relevant open questions still remain to be addressed in the future. First, to our knowledge no other initial configuration other than the dumbbell has been proven to have a lower bound of  $\Omega(n^2)$ . Further work should investigate whether the dumbbell is a singular such configuration or if there are others. Second, future work should analyze the dynamics of clustering in configurations in order to rigorously address the  $2R$ -conjecture. Finally, for physical connectivity in the graph, future research should work to understand other classes of graphs that have bounded  $\epsilon$ -convergence times.

## References

- [1] Jan Haskovec. Well posedness and asymptotic consensus in the hegselmann-krause model with finite speed of information propagation. *Proceedings of the American Mathematical Society*, 149(08):3425–3437, 2021.
- [2] Rainer Hegselmann, Ulrich Krause, et al. Opinion dynamics and bounded confidence models, analysis, and simulation. *Journal of artificial societies and social simulation*, 5(3), 2002.
- [3] Arnab Bhattacharyya, Mark Braverman, Bernard Chazelle, and Huy L Nguyen. On the convergence of the hegselmann-krause system. In *Proceedings of the 4th conference on Innovations in Theoretical Computer Science*, pages 61–66, 2013.
- [4] Rohit Parasn timer, Massimo Franceschetti, and Behrouz Touri. Hegerselmann-krause dynamics with limited connectivity. In *2018 IEEE Conference on Decision and Control (CDC)*, pages 5364–5369. IEEE, 2018.
- [5] Sonia Martínez, Francesco Bullo, Jorge Cortés, and Emilio Frazzoli. On synchronous robotic networks—part ii: Time complexity of rendezvous and deployment algorithms. *IEEE Transactions on Automatic Control*, 52(12):2214–2226, 2007.
- [6] Edvin Wedin and Peter Hegarty. A quadratic lower bound for the convergence rate in the one-dimensional hegselmann–krause bounded confidence dynamics. *Discrete & Computational Geometry*, 53(2):478–486, 2015.
- [7] Peter Hegarty and Edvin Wedin. The hegselmann-krause dynamics for equally spaced agents. *Journal of Difference Equations and Applications*, 22(11):1621–1645, 2016.
- [8] Edvin Wedin. *Results on the hegselmann-krause model in opinion dynamics*. PhD thesis, Master dissertation, 2014.
- [9] Vincent D Blondel, Julien M Hendrickx, and John N Tsitsiklis. On the 2r conjecture for multi-agent systems. In *2007 European Control Conference (ECC)*, pages 874–881. IEEE, 2007.
- [10] Ravi Kannan, Santosh Vempala, and Adrian Vetta. On clusterings: Good, bad and spectral. *Journal of the ACM (JACM)*, 51(3):497–515, 2004.
- [11] Rohit Parasn timer, Massimo Franceschetti, and Behrouz Touri. On the convergence properties of social hegselmann-krause dynamics. *IEEE Transactions on Automatic Control*, 2021.
- [12] Rohit Parasn timer, Massimo Franceschetti, and Behrouz Touri. On graphs with bounded and unbounded convergence times in social hegselmann-krause dynamics. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, pages 6431–6436. IEEE, 2019.