Consensus in Complex Networks, with Application to Flocking

Harvey Devereux

May 1, 2018

1 Introduction

1.1 Outline

The idea of *consensus* in the context of a network of agents has broad applications including flocking and swarming, synchronization of coupled oscillators, dynamic graphs and control of multi-robot systems. The common aspect of consensus problems is for all agents in a system to agree on the value of some decision variable, an example is the problem of velocity matching within a flock.

The scope of this essay is to produce an introduction to the theory of consensus problems on networks of agents found in recent literature, and to discuss the relationship between the convergence of consensus algorithms and the structure of fixed and dynamic networks upon which they operate. The ideas presented are first investigated numerically for two important network models to characterise this struture-convergence relationship, and then dicsussed in the context of flocking alogrithms and their numerical simulation.

1.2 Literature

This essay in the most part considers three papers. The first by R. Olfati-Saber et al [1], considers consensus algorithms in full generality and mainly presents theoretical results on the convergence properties of consensus algorithms and concludes with a brief numerical investigation that is built on in section 2 of this essay. The second paper by H.G. Tanner et al [2] presents a simple flocking algorithm mainly for the purpose of investigating consensus (specifically velocity matching) rather than producing a realistic model of flocking, in this context it serves as a useful bridge between the more abstract nature of [1] and flocking. Finally [3], also by R. Olfati-Saber focuses mostly on constructing and analysing realistic flocking algorithms which go as far as object avoidance.

1.3 Consensus

In general a consensus problem on a network can be represented by a directed or undirected graph where, each agent is represented by a vertex in the graph and a decision parameter, and the agents interact over the graph topology with their neighbours [1].

That is we consider a graph $G = (V, E)$[1] where $V \in \{1, 2, \dots, n\}$ is the set of all vertices and $E \subseteq V \times V$ is the set of all edges, so each $v \in V$ is an agent and each $e \in E$ representing an interaction. Then associated with each agent $v \in V$ is a decision parameter $x_v \in \mathbb{R}^d$, for $d \in \mathbb{N}$, here we take $d = 1$ and drop the d unless otherwise noted. So the unconstrained[2] consensus problem can be represented using the decision vector $\boldsymbol{x} \in \mathbb{R}^{|V|} = [x_1, x_2, \dots, x_{|V|}]^T$ and we may define the notion of agreement as

$$x_1 = x_2 = \cdots = x_{|V|} \quad \text{or} \quad \boldsymbol{x} = \alpha \mathbf{1} \tag{1}$$

where $\alpha \in \mathbb{R}$ and $\mathbf{1} = [1, 1, \dots, 1]^T$. It is useful to define a set of neighours of agent i (those agents which influence i) as follows

$$N_i \triangleq \{j \in V : (i, j) \in E\} = \{j \in V : \boldsymbol{A}_{i,j} \neq 0\} \tag{2}$$

where $\boldsymbol{A} \in \mathbb{R}^{|V||V|}$ is the adjacency matrix of the graph G with non-zero entries, $\boldsymbol{A}_{i,j}$, if and only if $(i, j) \in E$. Finally note That the interation topology, G, may also be time-varying (dynamic) or perhaps even weighted to reflect the strength of influence between agents. Models with dynamic and fixed topology will be discussed below.

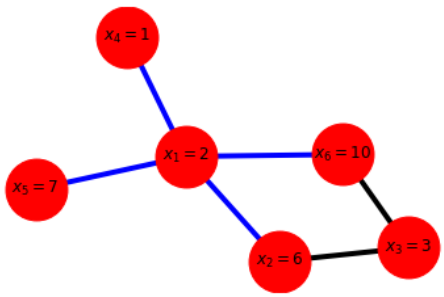

Figure 1: Example interaction topology with decision parameters, and N_1 connnections drawn in blue.

1.3.1 Fixed Topology Consensus Algorithms

With a fixed topology G representing agent interactions and a decision vector $\boldsymbol{x}$ as define above a consensus algorithm takes the form of some dynamics $\dot{x}_i$ for each agent i in the network. The first algorithm is a simple *average consensus* [1] model (also known as the distributed consensus algorithm) defined on an undirected graph G with adjacency matrix

[1] Throughout this essay we assume all graphs $G = (V, E)$ have no self edges or multiple edges, i.e $(i, i) \notin E$ and $(i, j) \in E$ is unique.

[2] A constrained consensus problem (or f-consensus problem) involves all agents compute a function $f(x)$ and must all asymptotically converge to $f(z)$ [1]

$\boldsymbol{A}$ as follows

$$\dot{x_i}(t) = \sum_{j \in N_i} \boldsymbol{A}_{i,j}(x_j(t) - x_i(t)) \tag{A1}$$

by defining the Laplacian matrix of the graph $\boldsymbol{L} \triangleq \boldsymbol{D} - \boldsymbol{A}$, where $\boldsymbol{D}$ is the diagonal matrix with $\boldsymbol{D}_{ii} \triangleq$ degree of vertex i. Using the undirected nature of G and the fact $\boldsymbol{A}_{i,j} = 0$ if $j \notin N_i$, it follows that

$$\begin{aligned}(-\boldsymbol{L}\boldsymbol{x})_i &= \sum_j \Big(\boldsymbol{A}_{i,j}x_j - \boldsymbol{D}_{i,j}x_j \Big) \\ &= \sum_{j \in N_i} \boldsymbol{A}_{i,j}x_j - \boldsymbol{D}_{ii}x_i \\ &= \sum_{j \in N_i} \boldsymbol{A}_{i,j}(x_j - x_i)\end{aligned}$$

hence algorithm A1 may be re-written as

$$\dot{\boldsymbol{x}} = -\boldsymbol{L}\boldsymbol{x} \tag{3}$$

and further by defining $\phi(\boldsymbol{x}) \triangleq \frac{1}{2}\boldsymbol{x}^T\boldsymbol{L}\boldsymbol{x}$ it is also to write algorithm A1 as the gradient descent algorithm on ϕ

$$\dot{\boldsymbol{x}} = -\nabla\phi(\boldsymbol{x}) \tag{4}$$

using equation 4 it is possible to show that algorithm A1 converges to average consensus when the graph G is connected [1], that is the solution to the consensus problem is uniquely

$$\boldsymbol{x} = \alpha\boldsymbol{1}, \;\; \alpha = \frac{1}{|V|}\sum_i \boldsymbol{x}_i(0) \tag{5}$$

One final property of algorithm A1 is that the speed of convergence is closely related to the algebraic connectivity of the graph G such that the rate of convergence is faster or equal to this number [1]. The algebraic connectivity of a graph is defined as the second smallest eigenvalue $\lambda_2 > 0$ of the Laplacian matrix $\boldsymbol{L}$.

A simple but general extension to algorithm A1 is to apply a weight γ_i to each agent i. This can be formalised by writing

$$\boldsymbol{K}\dot{\boldsymbol{x}} = -\boldsymbol{L}\boldsymbol{x} \tag{A2}$$

where $\boldsymbol{K} = \mathrm{diag}(\gamma_1, \gamma_2, \ldots, \gamma_{|V|})$ is the matrix of weights for all the agents. The algorithm A2 may also be written in the summation form as

$$\gamma_i\dot{\boldsymbol{x}}_i = \sum_{j \in N_i} \boldsymbol{A}_{i,j}(x_j - x_i)$$

and so interpreted as each node having a variable rate of integration. Note that A1 is a special case of A2 where the weight matrix is simply the identity matrix $\boldsymbol{I}_{|V|}$, and in general the system will converge to an weighted-average consensus of the form

$$\boldsymbol{x} = \alpha\boldsymbol{1}, \alpha = \frac{\sum_i \gamma_i x_i(0)}{\sum_i \gamma_i} \tag{6}$$

a specific example of A2 is the use of the weight matrix with elements $\gamma_i = |N_i|$ which has been used in the context of multi-vehicle formation control.

1.3.2 Fixed Topology with Communication Delays

A consensus algorithm may be extended to include a communication delay τ_i for each agent i, for example algorithm A1 with time delay may be written as

$$\dot{x}_i(t) = \sum_{j \in N_i} (x_j(t - \tau_{j,i}) - x_i(t - \tau_{i,j})) \tag{A3}$$

where in general $\tau_{i,j} = \tau_{j,i}$ need not be true and further the time delays may change in time. The simplest case is that of the uniform time delay where $\tau_{i,j} = \tau$ a constant for all $i, j \in V$ then algorithm A3 may be simplified to the form

$$\dot{\boldsymbol{x}} = -\boldsymbol{L}\boldsymbol{x}(t - \tau)$$

It was proven by Olfati-Saber and Murray (referenced in [1]) that algorithm A3 will solve an average consensus problem for all initial states if and only if the uniform time delay satisfies

$$0 \leq \tau < \frac{\pi}{2\lambda_n} \tag{7}$$

where λ_n is the largest eigenvalue of $\boldsymbol{L}$. It is also true that $0 \leq \tau < \frac{\pi}{4\Delta}$ is a sufficient condition for convergence of A3 since $\lambda_n \leq 2\Delta$, an interesting consequence of this is that graph structures that include one or many nodes with large degrees are more susceptible to time delays. For example the Barabási-Albert model generates a scale-free structure and hubs which are nodes with high degree [4][1].

1.3.3 Dynamic Topology

For a dynamic topology (switching network) we must consider a dynamic graph $G_{s(t)}$ where $s(t) : \mathbb{R} \mapsto J$ is a switching signal and J is an index set $J = \{1, 2, \ldots, m\}$. So we may represent algorithm A1 by the equation

$$\dot{\boldsymbol{x}} = -L(G_{s(t)})\boldsymbol{x} \tag{A4}$$

where we now consider $L(G_{s(t)}) : \Gamma \mapsto \mathbb{R}^{|V||V|}$ to represent the Laplacian of graphs $G_{s(t)} \in \Gamma$ where $\Gamma \triangleq \{G_1, G_2, \ldots, G_m\}$ is the set of possible topologies.

Algorithms of the form A4 converge to an average consensus whenever all graphs $G_k \in \Gamma$ are strongly connected, and either balanced digraphs or undirected graphs. The speed of this convergence is related to the second smallest eigen value of $\frac{L+L^T}{2}$, which is simply the algebraic connectivity for an undirected graph.

Altogether the algorithms A1, A2, A3, and A4 already provide simple models for a range of real conditions and are crucially all of these algorithms are highly dependent on the network structure in terms of the rate of convergence and whether the system converges to consensus at all. From these considerations a discussion of the structure of networks is relevant, as such the next section 1.4 considers two important network models and there properties.

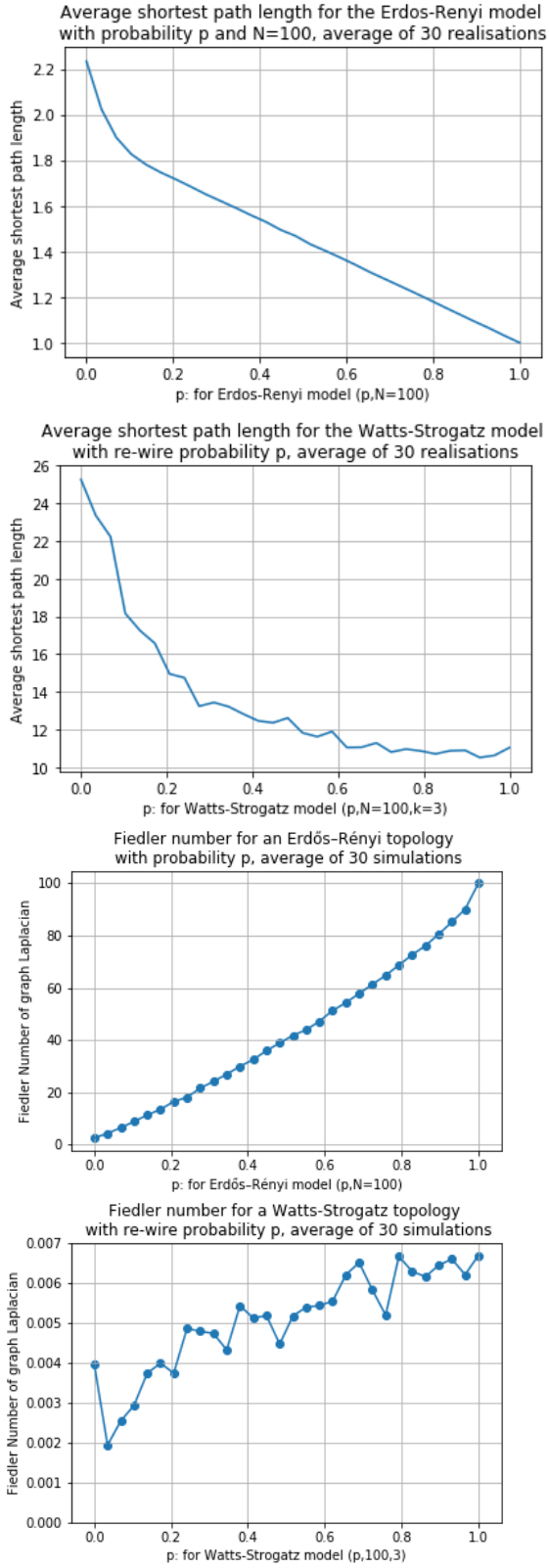

Figure 2: The Erdős-Rényi Model exhibits a lower average shortest path length and higher Fiedler number than the Watts-Strogatz model but both do show an decrease in average shortest path length and increase Fiedler number with p. Only connected graphs were considered.

1.4 Network Structure

Two random network models will be discussed in this essay in relation to the convergence of consensus algorithms, namely the Erdős-Rényi, and Watts-Strogatz models. Full details on all the models is found in [4].

1.4.1 Erdős-Rényi Model

An Erdős-Rényi random graph $G \sim \mathcal{G}^{ER}_{N,p}$ is a graph with N vertices, where each possible edge is present with probability $p \in [0,1]$. So the probability that the graph G has L links is given by

$$P_L = \binom{\frac{N(N-1)}{2}}{L} p^L (1-p)^{\frac{N(N-1)}{2}-L} \quad (8)$$

1.4.2 Watts-Strogatz Model

A Watts-Strogatz random graph $G \sim \mathcal{G}^{WS}_{N,k,p}$ is formed by a sequential re-wiring of a $2m$-regular ring graph R of size $|V| = N$ with $k = mN$ edges where vertices are connected if and only if $|i-j| \leq m$. The re-wiring process considers each vertex $i \in V$ of R in turn and for each edge $e = (i,j) \in E$ of R if $j > i$ with probability $p \in [0,1]$ e is replaced by the edge (i,l) where $l \in V$ is chosen uniformly at random among $v \notin N_i$.

1.4.3 Average Shortest Path Length and Fiedler Number

Both the Fiedler number (algebraic connectivity) and average shortest path length are measures of connectivity and efficiency of information flow across a network [4]. Therefore it is an intuitive proposition that these measures may predict the speed of convergence of a consensus algorithm on a given network structure, and as mentioned in section 1.3 the algebraic connectivity of a graph is a lower bound on the rate of convergence of consensus algorithms A1 and A4.

Intuitively the average shortest path length (or characteristic path length) is a measure of the expected number of steps required to walk between any two vertices in a graph. To define the average shortest path length formally first we define a path γ_{ij} as a sequence $\gamma_{ij} \triangleq (v_1 = i, v_2, \ldots, v_{l+1} = j)$ where each v_k is a vertex in the graph and $v_k \neq v_{k'}$ for all $k \neq k'$. Then $l = |\gamma_{ij}|$ is the path length of path γ_{ij} and the distance between two vertices i and j can be defined as $d_{ij} \triangleq \text{argmin}_l \gamma_{ij}$. If no path exists between two vertices then $d_{ij} \triangleq \infty$. Using this we can define the average path length of a graph G with N vertices as

$$PL(G) = \frac{1}{N(N-1)} \sum_{i,j \in V} d_{ij} \tag{9}$$

The algebraic connectivity of a graph is equal to the second smallest eigenvalue of the Laplacian matrix. Using the Gershgorin theorem it is implied that all eigenvalues lie in a disk of radius Δ being the maximum degree of the graph and further using the symmetry of the graph Laplacian L for an undirected graph it is possible to order the eigenvalues of L as follows [1].

$$0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n \leq 2\Delta \tag{10}$$

the Fiedler number λ_2 is only strictly greater than the trivial eigenvalue $\lambda_1 = 0$ if the underlying graph is connected. Figure 2 shows the average shortest path length as a function of p for $\mathcal{G}^{ER}_{N,p}$ and $\mathcal{G}^{WS}_{N,k=3,p}$ respectively, both generally decrease with p. In the Erdős-Rényi model this is quite obvious as the model tends to the complete graph $\mathcal{K}_N$ as $p \to 1$ since all edges are present so equation 6 is 1 only if $L = \frac{N(N-1)}{2}$ and 0 otherwise. It is important to notice that the Fiedler number changes little with p for Watts-Strogatz model and a fixed average degree k and is clearly of very small magnitude for $k = 3$ when compared to the Erdős–Rényi model. But when p is fixed and the average degree k is varied over the range seen naturally from the Erdős–Rényi model the same dependence of the Fiedler number on k (in Watts-Strogatz) is seen as p (in Erdős–Rényi). This is because the Fiedler number is actually dependent on the degree of the graph as shown in equation 8, so for a Watts-Strogatz model with $N = 100, k = 3, p = 1$ the bound on the *largest* eigenvalue is $\approx 2 \cdot 3 = 6$ but for Erdős–Rényi $N = 100, p = 1$ the bound is around 200 which explains the behavior seen in figures 2 and 4.

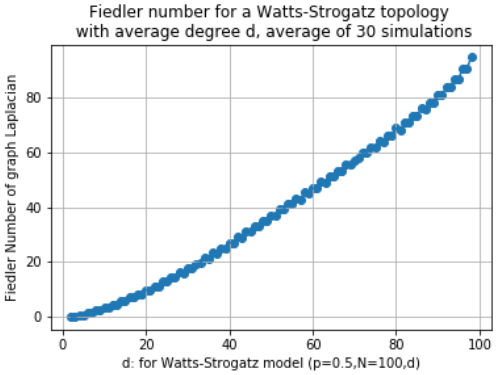

Figure 3: The dependence of the Fiedler number on the average degree passed to the Watts-Strogatz model is essentially identical to the dependence on probability in the Erdős–Rényi model.

1.5 Application to Flocking

Broadly flocking can be described by a collection of three heuristic rules known as Reynolds rules [5] these are

1 'Flock Centering: attempt to stay close to nearby flockmates'-(cohesion)

2 'Obstacle Avoidance: avoid collisions with nearby flockmates'-(separation)

3 'Velocity Matching: attempt to match velocity with nearby flockmates'-(alignment)

It is in the third rule that flocking becomes relevant in the study of consensus algorithms where we may formulate a consensus problem for a 2-dimensional decision parameter (the velocity of the agents). In fact many flocking algorithms may be decomposed into a 'gradient' based term (loosely encompassing rules 1 and 2) and a consensus term corresponding to 3. [3] emphasises this view point in particular.

Common to all flocking algorithms is the basic structure of the dynamics of the agents, we define the position and velocity of agent i to be $\boldsymbol{p}_i, \boldsymbol{v}_i \in \mathbb{R}^m$ respectively. The dimension m may be arbitrary, we shall take $m = 2$ unless otherwise noted. In general the agents will for a system

$$\begin{cases} \dot{\boldsymbol{p}}_i = \boldsymbol{v}_i \\ \dot{\boldsymbol{v}}_i = \boldsymbol{a}_i \end{cases} \tag{11}$$

where $\boldsymbol{a}_i : \mathbb{R}^m \mapsto \mathbb{R}^m$ is some acceleration function. As with general consensus problems we need a network to define the interaction between agents, the definition of this topology varies between authors two such choices will be discussed in this essay.

The first algorithm discussed, presented in [2], defines a velocity graph and a sensing graph. Considering a set of agents $V \triangleq \{1, 2, \ldots, N\}$ with positions $\boldsymbol{p}_i$ for $i = 1, 2, \ldots, N$ the velocity and sensing graph are defined as $G_v \triangleq (V, E_v \subseteq V \times V)$ and $G_s \triangleq (V, E_s)$ respectively where $E_s \triangleq \{(i, j) \in V \times V : ||\boldsymbol{p}_i - \boldsymbol{p}_j||_2 \leq R\}$[3]. Now defining the two sets of neighbours N_i^v, N_i^s for graphs G_v, G_s respectively the flocking algorithm is defined by

$$\boldsymbol{a}_i = -\sum_{j \in N_i^v} (\boldsymbol{v}_i - \boldsymbol{v}_j) - \sum_i \sum_{j \in N_i^s} \nabla_{\boldsymbol{p}_i} V_{ij}(||\boldsymbol{p}_i - \boldsymbol{p}_j||_2) \tag{F1}$$

where R is a spacing parameter and V_{ij} is an arbitrary non-zero and differentiable potential function satisfying

1 $V_{ij}(x) \to +\infty$ as $x \to 0$

2 $V_{ij}(x)$ has a unique minimum

3 $\frac{d}{dx} V_{ij}(x) = 0$ if $x > R$

Similarly to the general consensus algorithm the convergence of the algorithm F1 (all agent velocities match) is assured as long as the velocity and sensing graphs are connected, this applies also to the case where the velocity graph is dynamic [2].

[3] Furthermore the velocity graph G_v may also be switching in which case $G_v = G_{s(t)} \in \Gamma$ for some set of topologies Γ as discussed in section 1.3.3.

The second and third flocking algorithms, proposed in [3], involve only a sensing graph, G_s exactly as defined above. A new distinction is made between α- and γ-agents which are agents of the flock and virtual agents representing the flocks goal (path or destination) respectively. As a result the dynamics are split into α and γ systems where $\boldsymbol{v}_i^\alpha, \boldsymbol{p}_i^\alpha, \boldsymbol{a}_i^\alpha$ represent the velocity,position, and acceleration of the ith α-agent and similarly for the γ-agent (in this essay a single γ-agent is considered many is possible). Before defining the algorithms an unwieldy number of definitions is necessary, full details of these are found in [3]. First a 'σ-norm' (not a norm) is a map $\mathbb{R}^m \mapsto \mathbb{R}_{\geq 0}$ defined as

$$||z||_\sigma \triangleq \frac{1}{\epsilon}\big(\sqrt{1+\epsilon||z||_2^2} - 1\big) \tag{12}$$

where $\epsilon > 0$ and has gradient

$$\sigma_\epsilon(z) \triangleq \nabla||z||_\sigma = \frac{z}{1+\epsilon||z||_\sigma} \tag{13}$$

Secondly a 'bump function' $\rho_h(z)$ is defined as

$$\rho_h(z) \triangleq \begin{cases} 1 & z \in [0,h) \\ \frac{1}{2}\big(1+\cos\big(\pi\frac{(z-h)}{(1-h)}\big)\big) & z \in [h,1] \\ 0 & \text{otherwise} \end{cases} \tag{14}$$

Thirdly the spatial adjacency matrix $\boldsymbol{A}^s$ is defined by

$$\boldsymbol{A}_{ij}^s \triangleq \begin{cases} \rho_h\left(\frac{||\boldsymbol{p}_j - \boldsymbol{p}_i||_\sigma}{||R||_\sigma}\right) & j \neq i \\ 0 & j = i \end{cases} \tag{15}$$

Fourthly

$$\phi(z) \triangleq \frac{1}{2}\big((a+b)\sigma_1(z+c) + a - b\big), c = \frac{|a-b|}{\sqrt{4ab}} \tag{16}$$

$$\phi_\alpha(z) \triangleq \rho_h(z/||R||_\sigma)\phi(z - ||d||_\sigma), d \geq 0 \tag{17}$$

The significance of d is that this is the inter-agent spacing to which the flock converges if the below algorithms converge. Fifthly

$$\boldsymbol{n}_{ij} \triangleq \sigma_\epsilon(\boldsymbol{p}_j - \boldsymbol{p}_i) \tag{18}$$

Finally the algorithms are

$$\boldsymbol{a}_i^\alpha = f_i^\alpha = \sum_{j \in N_i^s} \phi_\alpha(||\boldsymbol{p}_j^\alpha - \boldsymbol{p}_i^\alpha||_\sigma)\boldsymbol{n}_{ij}^\alpha + \sum_{j \in N_i^s} \boldsymbol{A}_{ij}^s(\boldsymbol{v}_j^\alpha - \boldsymbol{v}_i^\alpha) \tag{F2}$$

$$\boldsymbol{a}_i^\alpha = f_i^\alpha - c_1(\boldsymbol{p}_i^\alpha - \boldsymbol{p}^\gamma) - c_2(\boldsymbol{v}_i^\alpha - \boldsymbol{v}^\gamma), \;\; c_1, c_2 > 0 \tag{F3}$$

2 Simulation Results and Discussion

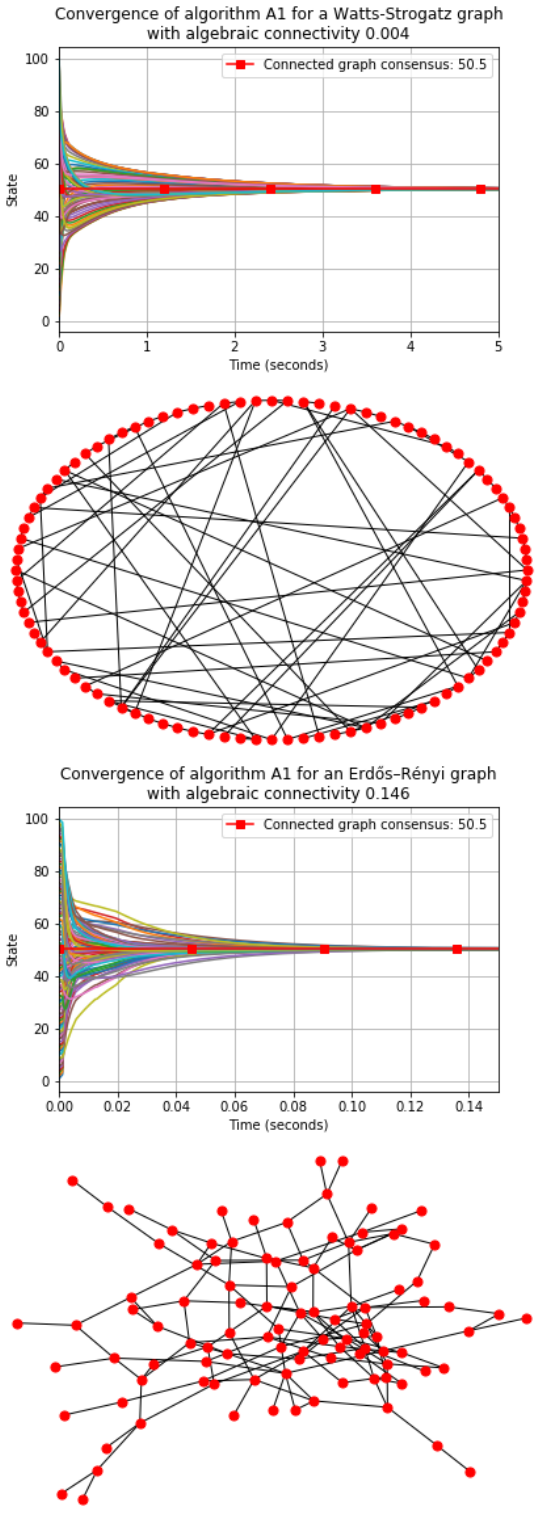

Figure 4: Convergence of A1 with the Watts-Strogatz network model $N = 100, k = 3$ and Erdős-Rényi network model $N = 100, p = 0.025$.

The following is a presentation of numerical simulations for the algorithms discussed above. Full implementation of the algorithms and some examples are given in [6].

2.1 Consensus Algorithms and Network Structure

2.1.1 Algorithm A1

Following from [1] algorithm A1 was simulated with a fixed topology generated from the Watts-Strogatz model with $N = 100$ and $k = 3$ and re-wire probability $p = 0.5$. Then algorithm A1 was simulated with topology generated from the Erdős-Rényi model. It is clear to see the large difference in convergence speed of the algorithm for these two topologies, and by examining the algebraic connectivities it can be estimated that the Erdős-Rényi topology give convergence at a rate $\approx \frac{0.146}{0.004} = 36.5$ times faster and indeed for agreement to 3 decimal places the convergence times are 0.145 and 5 making the Erdős-Rényi topology $\approx \frac{5}{0.146} = 34.5$ times faster. This is in good agreement with the convergence analysis [1].

The next simulations extend on by examining the behavior of the speed of convergence of algorithm A1 with the change in value of the probability parameter p in both the Watts-Strogatz and Erdős-Rényi models, the results of these simulation can be seen in figure 4 over the page. With the small anomaly of p close to 0 in the Watts-Strogatz model the behavior is perhaps expected given the graphs of the average shortest path length in section 1.4.3. We see a exponential decrease in simulation time as p increase, in the Erdős-Rényi model the simulation time is almost 0 as $p \to 1$ which is not so surprising given that the graph tends to a complete graph therefore all agents have knowledge of ev-

ery other agent (they interact with all other agents). Perhaps a little surprising is the performance of the algorithm with the Watts-Strogatz model which intuitively is designed for a short distance between any two given nodes (small world property). The key to understanding is the average degree which for the Watts-Strogatz model remains fixed but in the Erdős-Rényi the increases to $N-1$ (that of the complete graph) and for A1 being connected to as many vertices as possible insures better cooperation. Although the algorithm on the Watts-Strogatz model does generally perform much worse than the same algorithm on the Erdős-Rényi topology, there is still a notable decrease in runtime of roughly 70% this is perhaps explained by the change in average shortest path length which does decrease by about the same percentage when p is varied as shown in figure 2.

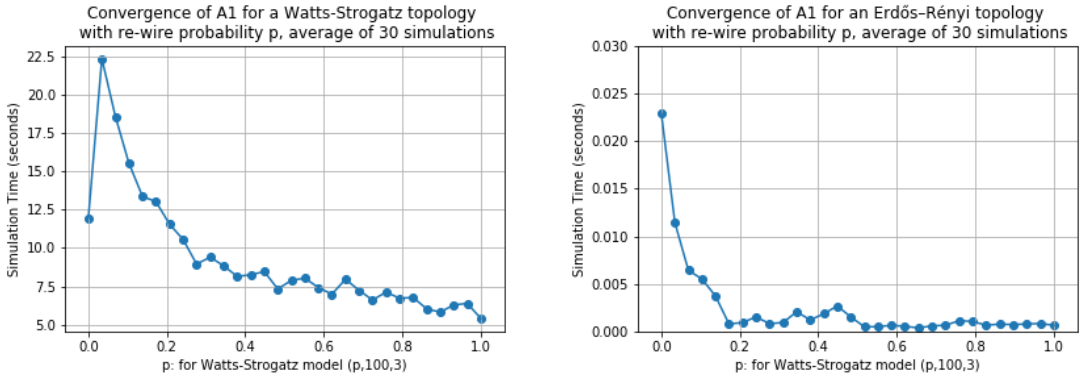

Figure 5: Speed of convergence of A1 for Erdős-Rényi and Watts-Strogatz models as a function of p. Only connected graphs considered.

The final simulation of involving A1 examined a randomly switching network where at each time step n double edge swaps were performed (where n is some proportion of the number of edges), the double edge swap itself takes edges $(x,y),(u,v)$ and replaces these edge with the new edges $(x,u),(y,v)$. This ensures the connectivity of the resultant graph and so A1 should converge regardless of the switching network. The result is shown in figure 5 below, clearly as the proportion of re-wire attempts increases there is a linear increase in the time taken to converge. This is interesting in comparison to both the fixed Watts-Strogatz topology with $p=1$, and also the same simulation with the Erdős-Rényi model in which no significant difference was seen.

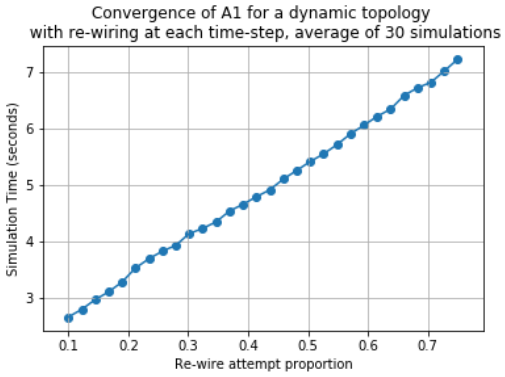

Figure 6: Speed of convergence for the algorithm A1 with a randomly switching topology. Initial topology is Watts-Strogatz $N=100, k=3,1$.

2.1.2 Weighted Average Consensus and Time Delays

The weighted average consensus algorithm A2 causes an interesting slow down in the rate at which consensus is reached as the magnitude of the weight matrix $\text{diag}(\gamma_1, \gamma_2, \ldots, \gamma_{|V|})$. This phenomenon can be seen in figure 7 where random weight vectors sampled uniformly from increasing intervals $[1, 2], [1, 3], \ldots, [1, 100]$ were used producing a clear trend.

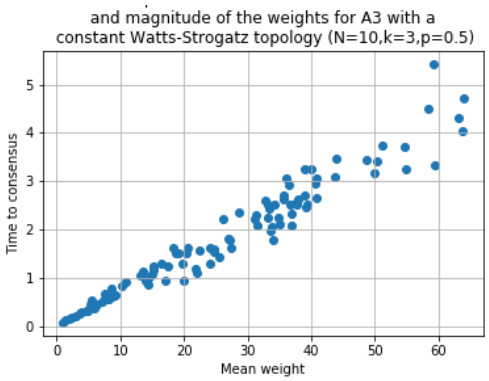

Figure 7: Time to consensus increase as the magnitude of the weights increases.

For the time delay model A3 a fixed graph with largest Laplacian eigen value λ_n such that $\tau_{max} \approx 0.29$ was used with delays of 90% and 103% to observe the effect of large time delay and a time delay just above the theoretical limit. The results show oscillatory behavior as seen in the literature ([1] for example).

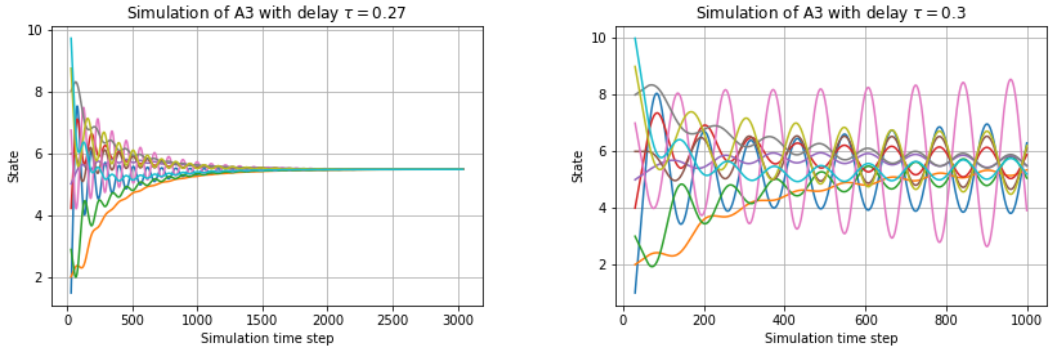

Figure 8: Simulation of A3 on a graph with $\tau_{max} \approx 0.29$ the algorithm clearly does converge with $\tau < \tau_{max}$ but with markedly more noise and oscillatory like behavior, this grows to full oscillations when the delay is larger than the maximum tolerance τ_{max}.

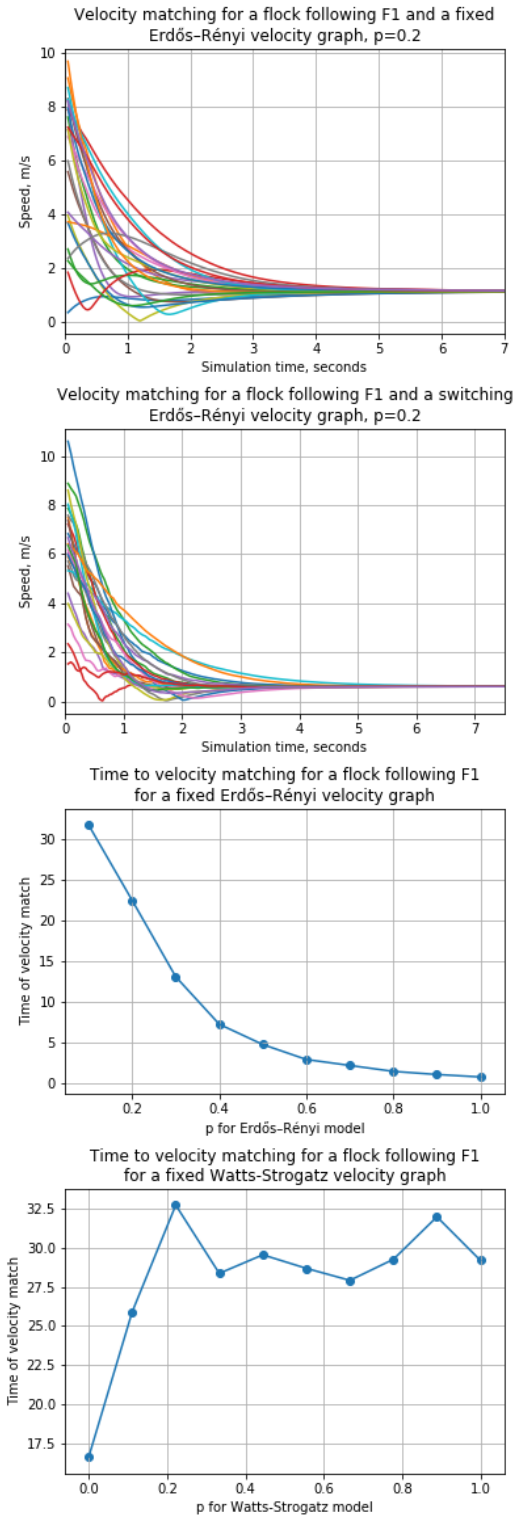

Figure 9: (top two) Velocity matching for flocks following F1 with a fixed and time varying topology. For the switching velocity graph at each time step the graph is randomly re-wired via a double edge swap just as in the final simulation of the previous section. (bottom two) dependence of time taken to converge to matched velocities for a flock following F1 and p for the Erdős–Rényi and Watts-Strogatz models.

2.2 Flocking Algorithms

2.2.1 Algorithm F1

The flocking algorithm F1 will result in flocking as long as the velocity graph is connected (or remains connected in the case of switching) so the behavior is very similar to A1. The evolution of each agents speed is shown in figure 9 for both fixed and switching velocity graphs. From these simulations it is interesting to observe that the switching of the network topology only served to cause slightly more noisy speed evolution of the flock members and altogether resulted in only a small increase in the time to convergence this is most likely due to the method of random switching used which has maintained the structure of the graph topology in terms of degree and so also in terms of the Fiedler number (to around 1 decimal place) which as discussed in section 1.4 determines the speed of convergence in a consensus algorithm and hence the speed of convergence to velocity matching.

Since the velocity graph with algorithm F1 is independent of the dynamics of the agents a similar analysis to figure 6 was performed to ascertain whether the speed of convergence to velocity matching has similar characteristics to A1. As can be seen in the bottom two images of figure 9 the time taken for the flock to converge to matched velocities does have similar dependence if the velocity graph is generated from the Erdős–Rényi model but this is interestingly not the case if the Watts-Strogatz model is used for the velocity graph where it appears the opposite relationship is observed.

2.2.2 Algorithm F2 and F3

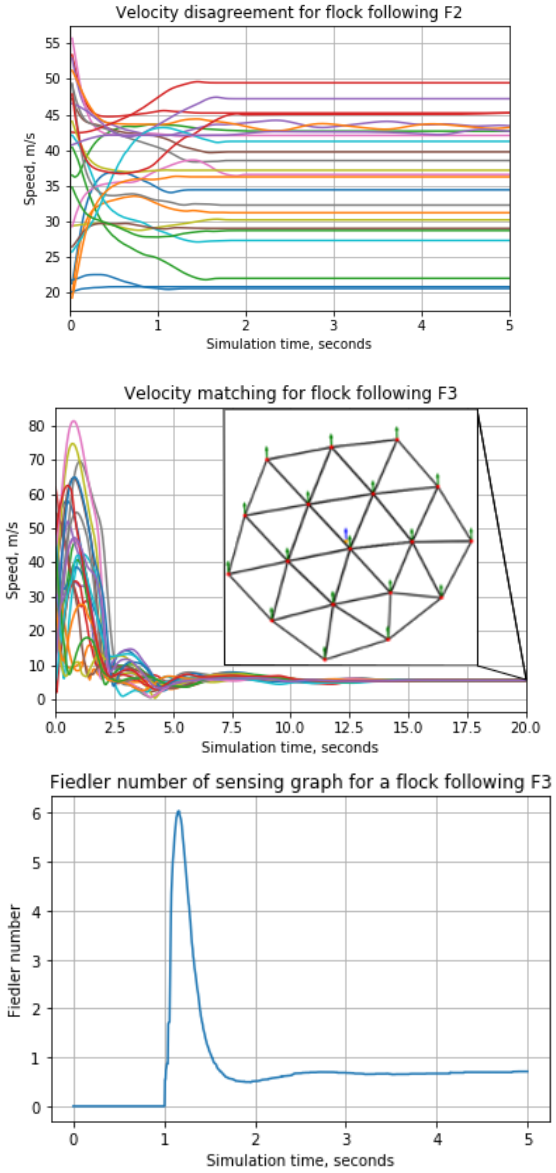

Figure 10: (top to middle) Velocity mismatching and matching for flocks following algorithms F2 and F3. The converged state of the flock following F3 is also embedded. The initial conditions of the flocks is randomised in the same way as the corresponding flocks in figure 11, and the γ-agent (blue in the embedding) was given a random initial position and random velocity which was kept constant throughout the simulation. Also pictured (bottom) is the angle agreement of the α-agents this plot shows the 'contraction-expansion' phenomenon where the flock moves together then away a little and so on until convergence.

Algorithms F2 and F3 are identical except for the presence of a γ agent which, represents the group objective. This difference culminates in ensured flocking for algorithm F3 and a very restricted set of initial conditions leading to flocking for algorithm F2 where regular fragmentation is most commonly seen even when the flock is started with almost identical speeds and positions [3].

For F2 the α-agent flock members initially behave in an erratic fashion (due to the random initialisation) which quickly evolves into a seemingly outward facing movement even though the sensing graph is still connected (CF figure 11 image 1,2), eventually this culminates in a situation identical to that of A1 evolving on a disconnected graph, that is we see most agents converging to a distinct velocity (no consensus is reached).

In the simulation of F3 we see a phase characterized by a large increase in velocity of the α-agents which represents the flock coming together and forming a group around the position of the γ-agent. A still of this phase can be seen in figure 11 images 5 which is opposite to the behavior of F2 in this middle phase. The agents then rapidly decrease speed to converge to the speed of the γ agent in a analogous way to A1 on a connected graph.

Although the network structure of the flock with algorithm F3 is inherently dynamic it is possible to see an interesting correspondence between the Fiedler number of the sensing graph and the evolution of the α-agents velocities, this is not surprising given the results on convergence for consensus algorithms on dynamic topologies, i.e A4.

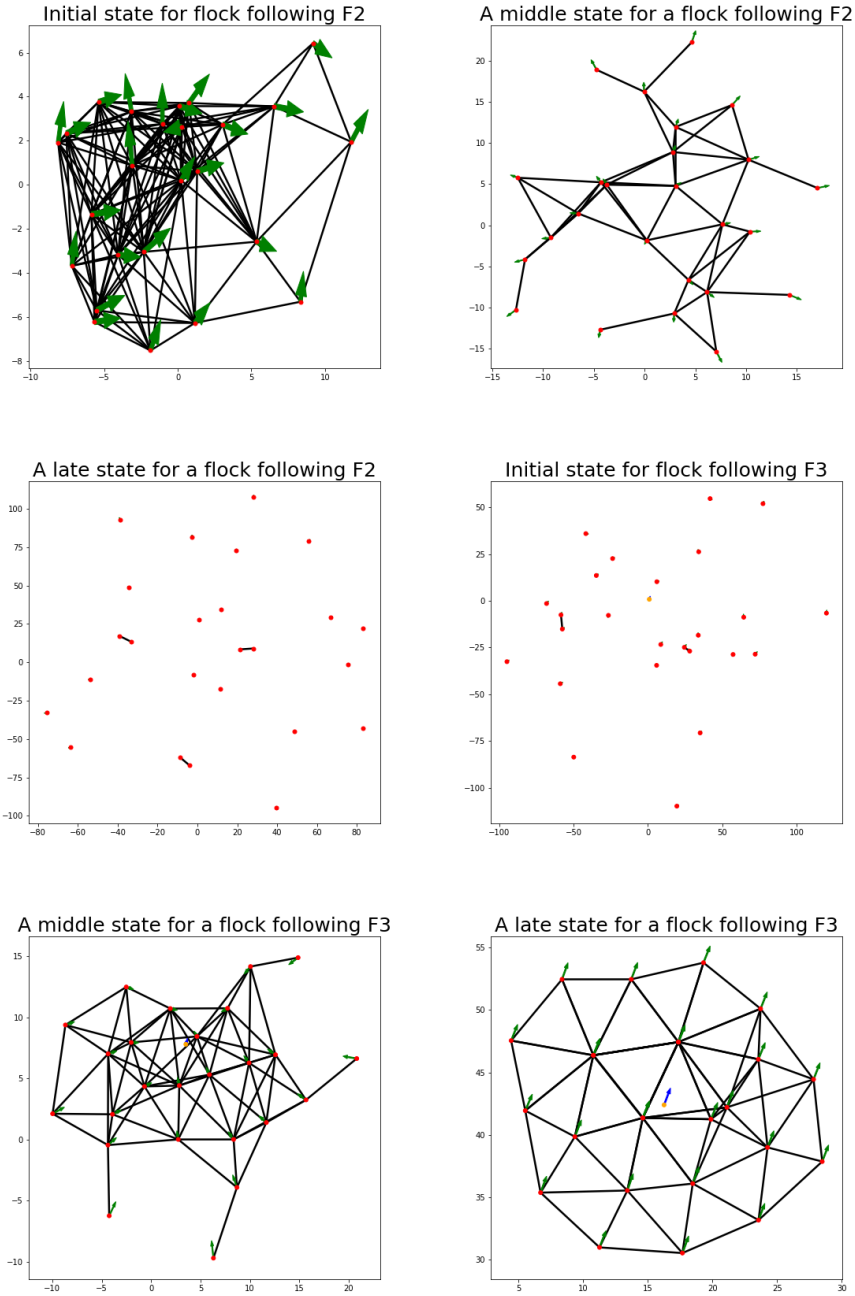

Figure 11: Stills showing initial, middle, and final (converged/diverged) states of a flock simulation using F2 and F3. Arrows represent velocity, red points are α-agents and a yellow point represents a γ-agent. The sensing network G_s is drawn in black. Note that F2 fails to form a stable flock even though the initial state is dense with high connectivity whilst F3 produces a stable flock even though the initial state is fragmented. For full animated gif of both simulations see the Readme.md of [6].

3 Conclusion

3.1 Consensus

The first paper discussed in this essay [1] provided an excellent overview of consensus algorithms and the role of network structure in their convergence properties since this is a more theoretical paper there is less emphasis is on numerical simulation with only a few cases discussed. In section 2 these results are extended slightly to investigate how the change of parameters in two random graph models effect the performance of the simple distributed consensus algorithm A1 and briefly the effect of switching with algorithm A4. Clearly as predicted by the theoretical results of R. Olfati-Saber et al the Fiedler number is crucial in evaluating the performance of a general consensus algorithm regardless of the network structure as the degree structure of the network changes, but as seen in the Watts-Strogatz model with fixed average degree (fixed k) and varying re-wire probability, there is still substantial decrease in the time taken to reach consensus even though the Fiedler number varies only between 0.002 and 0.007. In this situation other measure of information flow on networks such as the average shortest path length can potentially explain the convergence behavior.

3.2 Flocking

Consensus is clearly an important concept for flocking in general as per Reynold's rule 3, i.e the velocity matching concept. H. G. Tanner et al [2] presented a framework of two graphs one dynamic to handle flock member collision avoidance and a second fixed (or switching) network not based on agent dynamics to investigate the velocity matching, this is as mentioned by the authors not exactly a realistic model of flocking because considering a fixed or random interaction structure would imply the agents don't change which flock members they interact with for velocity matching or they choose at random respectively which does not seem like a physically or biologically realistic model. Nevertheless it is interesting to note that the convergence to velocity matching is not trivially identical to convergence of the general consensus algorithm A1 when considering a fixed velocity graph sampled via the Erdős-Rényi model and in particular the Watts-Strogatz model, that is the introduction of collision avoidance term of F1 (the first summation term being A1) has already introduced some interesting non-trivialities.

On the other hand R. Olfati-Saber et al [3] have produced a realistic-looking simulation of flocking via a perhaps quite artificial and complicated construction which is not so obviously connected to the underlying biology and physics of real-life flocking behavior. But even with the complicated structure of the models parts it is still true that consensus is only surely achieved with a connected graph and that the rate of convergence is related to the Fiedler number which implies a connection to the degree structure of the sensing network where a high average degree will imply a fast convergence.

3.3 Directions of Future Inquiry

Two potential directions of inquiry seem to present themselves, firstly to empirically evaluate the network structure of real-life flocking to see if the general principles of convergence of consensus algorithms on networks apply in reality. Secondly an investigation into more realistic and complex models of flocking could be made in the context of consensus algorithms.

References

[1] R. Olfati-Saber, J. A. Fax, and R. M. Murray. "Consensus and Cooperation in Networked Multi-Agent Systems". In: *Proceedings of the IEEE* 95.1 (Jan. 2007), pp. 215–233. ISSN: 0018-9219. DOI: 10.1109/JPROC.2006.887293.

[2] H. G. Tanner, A. Jadbabaie, and G. J. Pappas. "Flocking in Fixed and Switching Networks". In: *IEEE Transactions on Automatic Control* 52.5 (May 2007), pp. 863–868. ISSN: 0018-9286. DOI: 10.1109/TAC.2007.895948.

[3] R. Olfati-Saber. "Flocking for multi-agent dynamic systems: algorithms and theory". In: *IEEE Transactions on Automatic Control* 51.3 (Mar. 2006), pp. 401–420. ISSN: 0018-9286. DOI: 10.1109/TAC.2005.864190.

[4] A.L. Barabási and M. Pósfai. *Network Science*. Cambridge University Press, 2016. ISBN: 9781107076266.

[5] Craig W. Reynolds. "Flocks, Herds and Schools: A Distributed Behavioral Model". In: *SIGGRAPH Comput. Graph.* 21.4 (Aug. 1987), pp. 25–34. ISSN: 0097-8930. DOI: 10.1145/37402.37406.

[6] Harvey Devereux. *Consensus (Git-hub Repository)*. URL: https://github.com/harveydevereux/Consensus.