

Topological Aspects of Greedy Self-Organization

Furqan Ahmed and Olav Tirkkonen

Dept. of Communications and Networking, Aalto University

P.O. Box 13000, FI-00076 Aalto, Finland

Email: firstname.lastname@aalto.fi

Abstract—We consider self-organization problems, where agents try to agree about the value of a configuration space variable. Problems of consensus and synchronization belong to this category. These are the problems which would often be trivial to solve in a centralized setting, and non-trivial aspects are often directly induced by the process of self-organization itself. We discuss topological reasons as to why simple locally greedy algorithms are not able to create long-range order. The reason why greedy synchronization of a real-valued variable works in a straight forward manner, whereas greedy phase synchronization does not, is topological; in the latter non-trivial homotopy classes in mappings from the interaction graph of the agents to the configuration space exist. We identify higher-dimensional configuration spaces with such non-trivial homotopy classes. However, we find that greedy self-organization is able to create long-range order for any higher-dimensional configuration space that does not have circular components.

I. INTRODUCTION

In self-organizing systems, the actions of agents are based on their interactions with their respective peers, depending on rules and policies guided by local information leading to an emergence of a global pattern [1]. Synchronization and consensus problems are some of the simplest self-organization problems, and offer a platform for studying different issues that arise from self-organization.

In consensus problems, agents aim at reaching an agreement on the values of variables. It is a fundamental problem in areas of distributed computing and complex networks and therefore, has been of historical research interest [2], [3]. It has wide-ranging applications in the practical self-organizing systems. These include wireless sensor networks, unmanned air vehicles, air-traffic control—to name a few. A closely related problem is the network synchronization problem, in which a number of agents strive to achieve synchrony in their respective clocks. Mutual event synchronization is a special case of network synchronization, where nodes are peers and synchronize to each other. Existing approaches to tackle the problem are inspired by biological oscillators such as fireflies and pacemaker cells of heart. Most studies on synchronization, or consensus, have been either for one-dimensional configurations (real variables, or phases), or for Euclidean spaces. The effects of the communication topology on the convergence of such algorithms was studied in [4]. Here, in addition, we study more generic problems, where the variable to be agreed upon takes values in a higher-dimensional sphere. This has been addressed in [5], [6], and for more generic Riemann manifolds in [7], [8]. This general set of problems has relevance to swarm dynamics [5], synchronization of satellite orbits [7], or agreeing about postures in photos [9]. The problem would also be interesting for a set of wireless receivers to agree about

the best transmission method of a multiantenna transmitter. The parameter space of a two-element transmitter can be understood as a 2D sphere, see e.g. [10].

A major challenge lies in addressing these problems in a self-organized manner, where each agent has a limited knowledge of the network. Such considerations are relevant to the practical aspects of real world systems, such as scalability and minimization of overhead related to message passing. In a centralized setting, both synchronization and consensus problems are straight forward to solve, the centralized controller may collect the information, and directly dictate the result. This is in contrast to distributed schemes that hinge upon limited cooperation based on local interactions among the neighboring nodes to minimize the amount of message passing in network. It is worth noting that under such conditions, convergence to a global optimum is not guaranteed, and is even impossible in certain cases especially when agents behave in a greedy way. From this perspective, these problems are particularly appealing—as the centralized version is trivial, all problems with finding a solution stem from the self-organization principle itself. This phenomenon can be explained by studying the structure of the interaction graph of the agents, which reveals the dynamics of interaction between them. Thus, the underlying topology that ensues when agents act in a self-organized manner to reach a solution, determines convergence characteristics of greedy self-organization. In particular, existence of cycles in the interaction may result in an impediment to the convergence.

In this paper, we focus on the convergence aspects of self-organization of consensus and synchronization problems. The topological aspects that obstruct the convergence to a global optimum in self-organizing systems are highlighted, followed by a discussion on how the change in topology can help in avoiding them. We start by analyzing one-dimensional systems. For cycle graphs, we identify multiple families of fixed points for both best response and gradient descent based self-organizing algorithms. Next we turn to higher-dimensional configuration spaces. We observe that when the network is a planar graph, and the configuration space is the surface of 2D sphere, it is possible that the graph wraps non-trivially around the configuration space. However, we show that when the power of the cost of disagreement between two neighbors is $p \leq 2$, greedy self-organizing algorithms will be able to order the network. However, for cost functions with a larger power in the cost of disagreement, network non-aligned network configurations that are fixed points of self-organization algorithms may exist.

The rest of paper is organized as follows: Section II describes the system model and formulation of the greedy

self-organization problem for different topologies. Section III considers one-dimensional configuration spaces—the real line, and the circle. Section IV discusses the higher dimension configuration spaces, showing simulation results for a numerical example from self-organizing network, as well as an approximative analysis. We conclude in Section V.

II. SYSTEM MODEL: GREEDY SELF-ORGANIZATION & CONSENSUS

Consider a multiagent system comprising of a set \mathcal{V} of nodes. Each agent communicates with a subset of the nodes with fixed and static topology, represented by a graph $G(\mathcal{V}, \mathcal{E})$, where \mathcal{E} is the set of edges, i.e. communicating pairs of nodes. The adjacency matrix of this graph is A . For simplicity we assume bidirectional non-weighted communications, so that the adjacency matrix elements $a_{ij} \in \{0, 1\}$ are symmetric. We define the neighborhood of node i as $\mathcal{N}_i = \{j \in \mathcal{V} \mid a_{ij} = 1\}$. Each node i has an opinion of a variable \mathbf{x}_i taking values in the node-configuration space \mathcal{M} . The network configuration is represented by the collection of variables $\mathbf{X} = [\mathbf{x}_i]_{i \in \mathcal{V}}$, which may be interpreted as a matrix, or a vector, depending on the dimensionality of \mathcal{M} .

Without loss of generality, we focus on the self-organized network synchronization problems, where the objective of the agents is to agree about this value: $\mathbf{x}_i = \hat{\mathbf{x}}, \forall i$. For graphs with the same in-degree and out-degree, the greedy algorithms considered here can be easily generalized to consensus algorithms, see [2], [5]. In consensus not only agreement matters, but the quality of the agreement is also important. Thus, the agreement $\hat{\mathbf{x}}$ should represent the set of individual initial opinions as well as possible, subject to a suitable norm. To understand the difference between synchronization and consensus, a couple of examples from wireless communications are in order. In synchronization, the objective is that the nodes agree upon the value of a variable, but it does not matter what precise value is agreed upon. As an example, consider frequency synchronization, where each node has an opinion of a real-valued clock frequency x_i , a parameter related to the local oscillator. The aim is to find a common understanding on the carrier frequency used for communication. From the perspective of communication it does not matter, which precise value \hat{x} is agreed upon. It may be one of the original x_i , or some other value. As an consensus problem [2], one may consider a protocol where the nodes change their communication pattern based on a perceived average network load. Each node has an opinion x_i of the local network load, and the task of the network is to reach consensus regarding the average load. In this case, it is important that all nodes agree on \hat{x} , but in addition, the value of \hat{x} should represent the average of x_i .

To address synchronization in a self-organized greedy manner, we define a local cost function for all $i \in \mathcal{V}$

$$c_i(\mathbf{X}) \triangleq \frac{1}{2} \sum_{j \in \mathcal{N}_i} \|\mathbf{x}_i - \mathbf{x}_j\|^p, \quad (1)$$

and the total network cost is

$$C(\mathbf{X}) = \sum_i c_i(\mathbf{X}) = \frac{1}{2} \sum_{(i,j) \in \mathcal{E}} \|\mathbf{x}_i - \mathbf{x}_j\|^p. \quad (2)$$

Here $\|\bullet\|$ is a distance norm on \mathcal{M} , and the exponent $p > 0$ indicates how large and small disagreements are compared. A

typical value would be $p = 2$, indicating that node i minimizes the mean square distance to its neighbors. It is straight-forward to show that if the nodes are interpreted as particles in a swarm, if c_i is interpreted as a potential energy, and kinetic energy is as usual, the particles follow the integrator dynamics studied in [5]. Here we are not interested in the particulars of swarm dynamics. We consider the simplest and most elementary greedy dynamics for reaching synchronicity, or consensus. We will see that in many cases, topological properties prevent convergence of these algorithms. This is the reason why more involved phase-locking algorithms are necessary, especially in the case of event synchronization [11]. For simplicity we shall consider a greedy algorithm, where each node tries to directly minimize (1). Note that in [5], a global order parameter was constructed:

$$\rho = \left\| \sum_i \mathbf{x}_i \right\| \quad (3)$$

and a Lyapunov potential $V = 1 - \rho^2$ was considered. Using this, convergence to an aligned state could be proved when the communication graph was fully connected. For this graph, (2) equals this Lyapunov potential.

In this system model, there are two objects of interest which may have a non-trivial topology. First, for synchronization and consensus, the node-configuration space is a smooth manifold, which has a set of topological properties such as the number of connected components, openness/compactness. We are particularly interested in the homotopy structure \mathcal{M} . Second, we have the graph $G(\mathcal{V}, \mathcal{E})$, which may have a trivial topology, or may have one or more cycles. Any graph can be proven to be topologically equivalent to a so-called wedge sum of multiple circles, i.e., a the space acquired when gluing together a number of circles at one point [12]. The aim is to investigate potential non-trivial phenomena arising in self-organization problems, that are caused by non-trivial interactions of these two topologies. Any network configuration \mathbf{X} gives rise to a mapping

$$M(\mathbf{X}) : G(\mathcal{V}, \mathcal{E}) \mapsto \mathcal{M} \quad (4)$$

where not only the vertices, but also the edges of the communication graph are mapped to \mathcal{M} . Non-trivial properties of this mapping may prevent emergence of global order.

III. ONE-DIMENSIONAL EXAMPLES

A. Mutual Synchronization: Real Line Topology

We first consider the simplest configuration space topology, where the variable at each node $i \in \mathcal{V}$ is a real number $x_i \in \mathbb{R}$. Thus, the cost of non-agreement for node i with neighborhood \mathcal{N}_i is

$$c_i(\mathbf{x}) = \frac{1}{2} \sum_{j \in \mathcal{N}_i} (x_j - x_i)^p. \quad (5)$$

The aim is to minimize the network cost. As this is a simple optimization over the real line, it can be solved optimally by employing *best-response* updates. The best response update constitutes selfish behavior where each node acts in a non-cooperative way and attempts to minimize its cost function responding to the strategies of other (neighboring) nodes. For node $i \in \mathcal{V}$, the best response update is given by

$$x_i^* = \arg \min_x \frac{1}{2} \sum_{j \in \mathcal{N}_i} (x_j - x)^2 \quad (6)$$

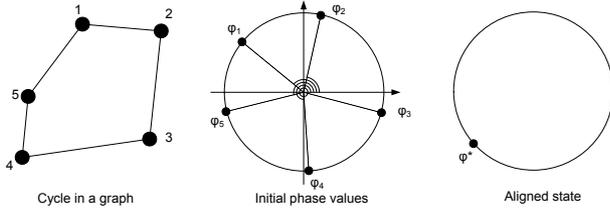


Fig. 1. Event synchronization can be mapped to phases on a circle.

Proposition 1. *For a real line topology, the best response update converges to an aligned state $x_i = x^* \forall i$.*

Proof. Consider a fixed point \mathbf{x}^* of the best response update scheme. As the configuration space is the real line, there is a unique order of the variables x_i , and thus there is a largest and a smallest value. If there is a fixed point such that the maximum and minimum are not the same, and the network graph is connected, at least one of the nodes having a maximum value has a neighbor with a smaller value. The best response of this node is smaller than the maximum value. Thus the maximum and minimum are the same in a fixed point. This proves that any fixed point is aligned:

$$x_i = x^* \quad \forall i \in \mathcal{V}. \quad (7)$$

Furthermore, a Cauchy sequence argument can be used to prove convergence to such a fixed point with probability one.

B. Mutual Event Synchronization: Circular Topology

Event-synchronization is a widely studied self-organization problem. The configuration space is one-dimensional, and with a non-trivial topology—the node configuration space \mathcal{M} is the circle S^1 . It is well-known that consensus and synchronization problems on the circle may not converge to an agreement, see e.g. the recent paper [8]. Here we shall enumerate all fixed points of greedy self-organization algorithms on a circle.

The variables on the circle can for example be represented by phases, $\phi_i \in [0, 2\pi]$. The vector of phases for the whole graph is denoted by Φ . In Fig. 1, a cycle graph, and its mapping to a circle is represented. Many distances can be defined on the circle, and as observed in [13], details of convergence performance may depend on the norm chosen, but the main principles are essentially the same. For simplicity, we here consider the *geodesic distance*, which is intrinsic to the manifold. The distance between two points ϕ_1 and ϕ_2 on the circle can be calculated along two paths, and the geodesic distance is the smaller. Thus we define

$$\|\phi_1 - \phi_2\|_G = \min(|\phi_1 - \phi_2|, |2\pi - \phi_1 + \phi_2|) \quad (8)$$

which can be regarded as timing difference or a measure of asynchrony between two nodes. The local cost due to non-alignment at a node is thus

$$c_i(\phi) = \frac{1}{2} \sum_{j \in \mathcal{N}_i} \|\phi_i - \phi_j\|_G^p. \quad (9)$$

However, in an aligned state, all phases are the same as illustrated in Fig. 1.

Algorithm 1 Synchronous best response self-organization

- 1: Initialize $\phi_i \in [0, 2\pi], \forall i \in \mathcal{V}$
- 2: At $t[n]$, all nodes $i \in \mathcal{V}$ update

$$\phi_i = \arg \min_{\phi} \sum_{j \in \mathcal{N}_i} \|\phi_j - \phi\|_G^p$$

- 3: Repeat until convergence or $n = \text{MaxIters}$.
-

C. Greedy Self-organization Algorithms

In greedy self-organization, each node tries to minimize its cost (9). Assuming that the nodes change their variables at discrete times, a straight-forward approach is that each node $i \in \mathcal{V}$ performs the following best-response update

$$\phi_i = \arg \min_{\phi} \sum_{j \in \mathcal{N}_i} \|\phi_j - \phi\|_G^p \quad (10)$$

To make the update formulation complete, we note that if the minimum is not unique, random selection is used. There are multiple alternatives related to the timing of the update of node i . If the variable ϕ represents node timing, as in true synchronization problems, the time of update may depend on ϕ_i . If ϕ represents a more general variable, updates may be synchronous or asynchronous.

Proposition 2. *With synchronous best response updates, the greedy algorithm for mutual event synchronization may not converge on a circular topology.*

Proof: At each instant t , due to selfish updates the total cost C_ϕ is not non-increasing. This results in oscillatory behaviour, as shown in Fig. 2, where all the nodes update their phases simultaneously by jumping on the mid-point of arc joining their respective neighbors.

In an asynchronous version, nodes would update their phases at unique time instants $t_i[n]$ in a given iteration. To this end, the synchronous and the asynchronous best response algorithms are summarized as **Algorithm 1** and **Algorithm 2**, respectively. A simpler algorithm for greedy self-organization, would be the local gradient descent. The gradient descent based update rule for node i is given by

$$\phi_i = [\phi_i + \beta \nabla_{\phi_i} C(\phi)]_{\mathcal{S}} \quad (11)$$

where $[\bullet]_{\mathcal{S}}$ is the projection on the feasible set $\mathcal{S} \triangleq [0, 2\pi]$, and $\nabla_{\phi_i} C(\phi)$ is the gradient with respect to ϕ_i given by

$$\nabla_{\phi_i} C(\phi) = \frac{\partial c_i(\phi)}{\partial \phi_i} + \sum_{j \in \mathcal{N}_i} \frac{\partial c_j(\phi)}{\partial \phi_i} \quad (12)$$

In the problem at hand, the cost of a node is perfectly aligned with the network cost, so no signaling is needed to devise gradient descent updates. The nodes simply take a small step to a direction where the cost is reduced. The updates in this case can also be synchronous or asynchronous. As opposed

Algorithm 2 Asynchronous best-response self-organization

- 1: Initialize $\phi_i \in [0, 2\pi], \forall i \in \mathcal{V}$
- 2: At $t_i[n]$, node $i \in \mathcal{V}$ updates

$$\phi_i = \arg \min_{\phi} \sum_{j \in \mathcal{N}_i} \|\phi_j - \phi\|_G^p$$

- 3: Repeat until convergence or $n = \text{MaxIters}$.
-

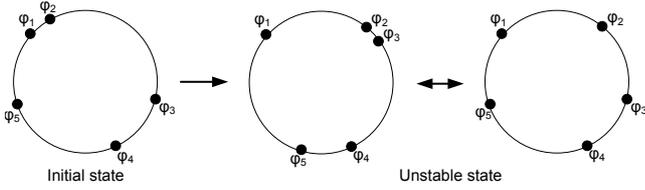


Fig. 2. Synchronous updates lead to unstable (oscillatory) behavior

to the best response algorithm, synchronous gradient descent will not be dramatically different from an asynchronous gradient descent, if the step size is small. The step-size governs how close to the minimum cost a fixed point could possibly be. A gradient descent with infinitesimal step size would give rise to a particle swarm dynamical system, of the type considered in [2].

D. Fixed Points of Cycle Graphs

The circular topology differs from the real line in that no unique order exists between phases. Accordingly, the best response and gradient descent have non-aligned fixed points, when the graph $G(\mathcal{V}, \mathcal{E})$ has cycles. For simplicity we consider a graph consisting of one cycle only, as depicted in Fig. 1. In this graph, each node has exactly two neighbors.

First, we note that if $p > 1$, at a fixed point Φ^* , the distances between two neighboring points are all the same. To see this, it is sufficient to observe that for a node with two neighbors, and $p > 1$, the function $c(\phi) = \sum_{j=1}^2 \|\phi - \phi_j\|_G^p$ has two local minima, the two midpoints on the two segments of the circle connecting the phases ϕ_1 and ϕ_2 of the two neighbors. One of these is the global minimum, found by the best response algorithm. This is at a distance $d \leq \pi/2$ from the two neighbors. The other is at a distance $\pi/2 \leq d < \pi$ from the neighbors. The sequence of points generated by gradient descent may be attracted to any of these two local minima. As a consequence of this, at a fixed point of the best response update, all ϕ_i are at the closest midpoint of its two neighbors, whereas for a gradient descent fixed point, it is sufficient for all ϕ_i to be at one of the two midpoints of its two neighbors. As the neighborhood relation on a cycle graph is cyclic, this leads to the following characterizations of fixed points.

Proposition 3. Consider a cycle graph with $I = |\mathcal{V}|$ nodes, and a network configuration characterized by phase variables $\{\phi_i\}_{i=1}^I$. The best response update algorithm has $\lceil \frac{I}{4} \rceil - 1$ families of fixed points, with configurations

$$\phi_i^* = i \frac{n2\pi}{I} + \varphi, \quad (13)$$

where n is an integer with $|n| < \frac{I}{4}$, and φ is a common phase. *Proof.* The difference between two consecutive phases is fixed, and $< \pi/2$. If the difference is exactly $\pi/2$, random breaking of evens in the best response algorithm will change the system configuration.

Proposition 4. Consider the same cycle graph. The gradient descent algorithm has $\lceil \frac{I}{2} \rceil - 1$ families of fixed points, with configurations of the form (13) where n is an integer with $|n| < \frac{I}{2}$.

Proof. Straight forward verification.

Note that in both cases, only the fixed point with $n = 0$ is an aligned state. If the initial points are selected randomly, each

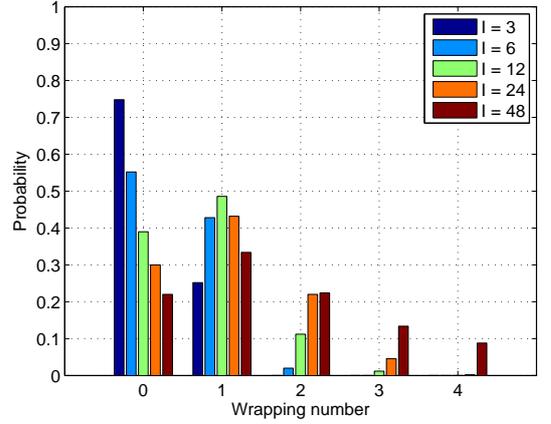


Fig. 3. Probability of convergence to fixed points with different wrapping number, for different number of nodes I . Asynchronous gradient descent updates.

of the possible fixed points characterized by n has an attraction pool of initial points with non-zero probability. This can be seen as follows. For the best response update, consider an angle $\alpha = \min(\frac{n\pi}{I}, \frac{\pi}{4} - \frac{n\pi}{I})$. Now any configuration with $i\frac{n2\pi}{I} - \alpha < \phi_i < i\frac{n2\pi}{I} + \alpha$ will converge to (13). This follows from the fact that with these initial conditions, the best response algorithm will not change the order of the ϕ_i , nor change any of the ϕ_i to the opposite side of the circle. For the gradient descent, a similar argument can be given.

Accordingly, any greedy self-organizing synchronization/consensus algorithm of a graph with cycles and circular configuration variables will face problems with converging to an aligned state. To proceed with synchronization, the symmetry between neighbors has to be broken. This is done in phase-locking algorithms [14], such as the classical firefly synchronization algorithm [11] by deliberately considering neighbors with slightly more advanced phase values more than neighbors with less advanced phase values.

Monte Carlo studies were performed studying the probability of the converged state to be a fixed point (13) that wraps around the configuration space n times. Results for asynchronous gradient descent are reported in Fig. 3, and results for asynchronous best response are reported in Fig. 4. The initial states consist of random phases, and the algorithms are run for $k = 200$ iterations. Convergence to fixed points with the largest wrapping numbers n is very improbable. However, convergence to fixed points with moderate wrapping numbers is rather probable, and as predicted more probable for the gradient descent than for best response updates.

E. Homotopy Classes in Circular Topology

The integer n in (13) characterizes the mapping $M(\Phi)$ of (4), i.e. how the graph maps to the node configuration space. When both the graph and the configuration space are topologically non-trivial, as in this case, this mapping may have different homotopy classes. Two mappings are homotopic, if one can be continuously deformed into the other. A collection of homotopic mappings is called a homotopic class.

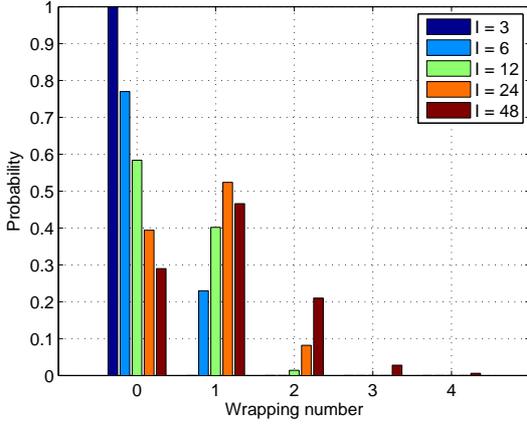


Fig. 4. Probability of convergence to fixed points with different wrapping number, for different number of nodes I . Asynchronous best response updates.

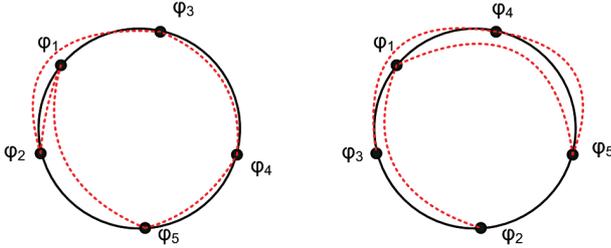


Fig. 5. Two mappings from a cycle graph to a circle, with wrapping number 0 and 1.

For the cycle graph and circular variables, the homotopy class n is characterizing how many times the graph wraps around the circle. Generically a circle may wrap around another circle infinitely many time, i.e. n may take any integer values. In the problems addressed here, the edges between the links are logical, however, and indicate a communication relationship. Thus an edge is always assumed to go between two neighbors along the shortest route. As a consequence of this, a cycle graph cannot wrap around a circle an infinite number of times.

Once the edges are drawn along the shortest path, we have a mapping of a circle to a circle. The wrapping number of this mapping can be easily calculated by calculating the *index* of a point in the configuration space [12]. With the configuration $\{\phi_i\}$, and the angles taking values between 0 and 2π , it is most straight forward to calculate the index of the point π . This is a signed sum of the number of edges that goes through the point π . If $\phi_i < \pi < \phi_{i+1}$, and $\phi_{i+1} - \phi_i < \pi$ this interval contributes +1 to the index, if $\phi_{i+1} < \pi < \phi_i$, and $\phi_i - \phi_{i+1} < \pi$, it contributes -1 to the index. It is easy to convince oneself that the wrapping number equals the index of any point. In Fig. 5, two configurations of a cycle graph are depicted, with the same five values of ϕ , but in a different order. The connections between neighbors in the graph are also depicted. One order gives rise to a mapping from the cycle to the circle with wrapping number 0, the other has wrapping number 1.

Gradient descent induces a continuous transformation of

this mapping of the graph to the circle, by smoothly moving the vertices of the graph, and accordingly stretching or shrinking the edges. Thus, according to elementary homotopy theory [12], the wrapping number is not changed by gradient descent. A topological interpretation of the best response algorithm is more delicate. In addition to changes where the updated point lies on the same circular segment between the two neighbors as the previous point, there is the possibility that the updated point jumps to the opposite line segment as shown in Fig. 6. The former can be interpreted as continuous deformations of the mapping from the graph to the circle. The latter, however, are discontinuous deformations, that change the homotopy class. However, once the best response algorithm has converged to a configuration where no jumps to the other side happen anymore, the best response updates preserve the wrapping number. Such configurations exist, and have a non-vanishing probability, as discussed in the context of Proposition 3. Note that in the gradient descent, no discontinuous changes of the homotopy class happen after initialization, a point moves always towards the midpoint of its two neighbors on the circle segment that it is situated on.

IV. HIGHER DIMENSIONAL CONFIGURATIONS

For one-dimensional configuration spaces, we observed that if the graph $G(\mathcal{V}, \mathcal{E})$ describing the communication topology has cycles, problems with aligning nodes may arise when the configuration space was circular, and thus had a non-trivial topology. It is an interesting problem to address, what happens in higher-dimensional configuration spaces. We consider smooth and homogeneous manifolds \mathcal{M} , in particular spheres and hyperspheres.

If the communication graph is literally interpreted as a topological collection of glued circles, the answer is clear. A mapping of a circle to a sphere or hypersphere can be continuously deformed to a point. The same holds for a collection of glued spheres. Pictorially speaking, each circle slips over the sphere, and can be contracted to a point. However, as we saw above, when discussing mappings of communication graphs to circles, the topological interpretation of the edges in communication graphs is not straight forward. The problem thus merits a deeper study. Some graphs, allow for another topological interpretation. If in addition to the

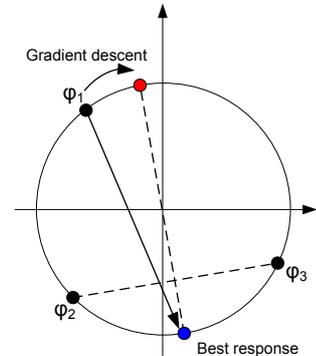


Fig. 6. Some best response updates may be seen as discontinuous transformations of the mapping of the cycle to the circle—the ones involving a jump to the opposite side. Gradient descent always gives rise to a continuous transformation of this mapping.

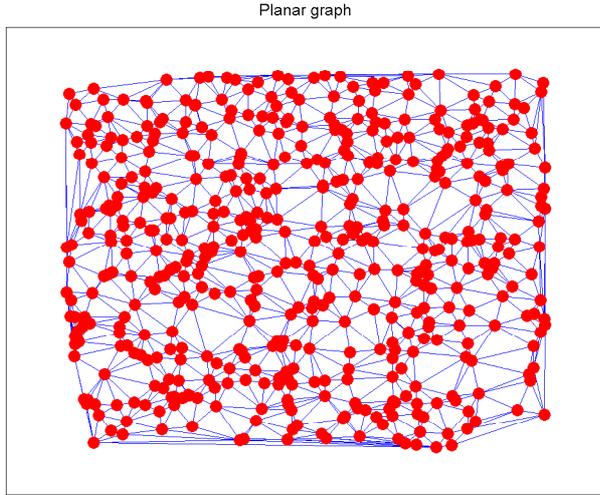


Fig. 7. Random planar Graph with $I = 500$ nodes.

vertices and edges, a natural geometrical definition of a face of a graph can be given, an Euler characteristic $\chi = V - E + F$ can be calculated for a graph, where V is the number of vertices, E the number of edges, and F the number of faces. In particular, any planar graph has $\chi = 2$, indicating that a planar graph can be understood as a discretization of S^2 , the two dimensional sphere in three Euclidean dimensions [12]. In this interpretation, the part of the 2D plane that is “outside” of the planar graph, is interpreted as a face, which is surrounded by the vertices on the perimeter of the graph, i.e. the area outside the graph itself is one face of the discretization of the sphere.

Figure. 7 shows a random planar graph, which can be understood as a discretization of a 2D sphere. It is created by dropping $I = 500$ nodes uniformly and at random in a rectangular area, performing a Voronoi tessellation of the area, and connecting nodes that share a boundary. If a planar graph is interpreted as a discretization of a sphere, and the configuration space \mathcal{M} is a sphere as well, mappings from the graph to \mathcal{M} may have a non-trivial homotopy class, characterized by a wrapping number n . It is not clear whether such wrapping numbers would have an effect on fixed points of greedy self-organization algorithms.

A. Planar Graph & Spherical Configurations

To get an understanding on the self-organization characteristics, we perform simulations on random planar graph, where the configuration space variables $\mathbf{x}_i \in \mathbb{R}^m$ are constrained to have unit norm. The planar graph is mapped on the sphere and nodes interact according to the neighborhood relations $\mathcal{N} - i$ determined by the graph. Thus when a node changes its variable \mathbf{x}_i , the node moves on the surface on sphere. With $p = 2$ in (1), a best response update is given by

$$\mathbf{x}_i^* = \frac{\sum_{j \in \mathcal{N}_i} \mathbf{x}_j}{\left\| \sum_{j \in \mathcal{N}_i} \mathbf{x}_j \right\|} \quad (14)$$

Here, the new value is the linear combination of the neighbor vectors, normalized to have unit norm.

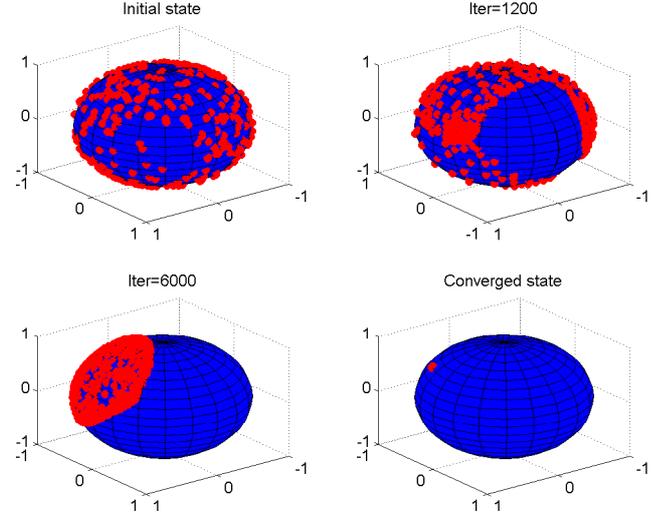


Fig. 8. Sphere Packing: Convergence to a fully aligned state with asynchronous gradient descent self-organization

To study configuration of self-organization on a random planar graph, we considered $\mathcal{M} \triangleq S^2$, i.e. the variables x_i at the nodes can be represented by e.g. real 3D unit norm vectors. This gives rise to a mapping of the planar graph to the 2D sphere. The resulting mapping or initial from a random selection of points x_i is shown in Fig. 8 for $I = 500$ nodes. Note that the edges in the communication graph are not drawn in the figure, and nodes that are close to each other on the sphere are not necessarily neighbors in the graph. The network is in complete disorder initially due to the random position of nodes on the sphere. The asynchronous gradient descent algorithm is run with step-size $\beta = 0.05$ and total iterations $k = 15000$. The snapshots at different iterations are captured as the network evolves to a completely ordered state. In Fig. 9 we depict the convergence towards an ordered state in a random planar graph with random initial points on the sphere, with different numbers I of nodes in the network. The power in the cost function is $p = 2$. A best response dynamic is used, and convergence is universal, and smooth.

As the random planar graph in this example has a very long perimeter, it is a bit dubious, how good a discretization of the sphere the graph is. There is one face, the face with the infinity point, which is surrounded by a very long cycle. To consider a more homogeneous discretization of a sphere, we consider a sphere packing. We take the best known packing with 32 points, from [15]. Thus each vertex i is characterized by a point \mathbf{v}_i on the circle. The communication topology is given by a nearest neighbor rule in the packing. As the objective is to understand whether the topology of the mapping from the graph to the sphere has a similar meaning as for the circle, we start with hand-crafted initial states where the planar graph wraps around the sphere. For this, we consider a configuration where each vertex i has exactly the configuration space variable corresponding to the vertex. The hand-crafted fixed point is unstable against any infinitesimal perturbation of the initial conditions, $\mathbf{x}_i = \mathbf{v}_i + \epsilon_i$. In this case, the initial condition, some intermediate states, and the convergence point of the best response dynamic (14) are depicted in Fig. 10. From this figure we see how all the points are gradually

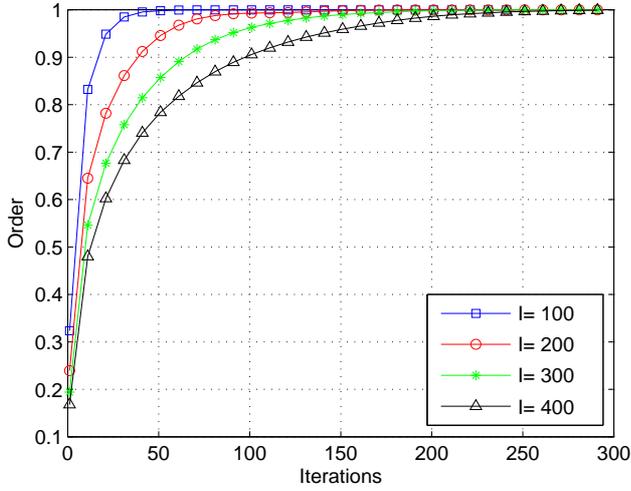


Fig. 9. Order parameter vs iteration

collected to one hemisphere, so that the sphere representing the configuration space squeezes through one of the faces of the planar graph. With the best response dynamic the density of the points on one hemisphere decreases, until suddenly all points jump to one hemisphere, and the convergence is rapid.

B. Approximative analysis

To understand the convergence of mappings of planar communication graphs to a 2D spherical configuration space, we focus on an approximative formulation of the underlying system dynamics, and concentrate on infinitesimal gradient descent with $p > 1$.

As discussed above for the circular case, a gradient descent is a continuous deformation of the mapping of the graph to the configuration space. In the situation of interest here, the points corresponding to the mapping of the vertices to the

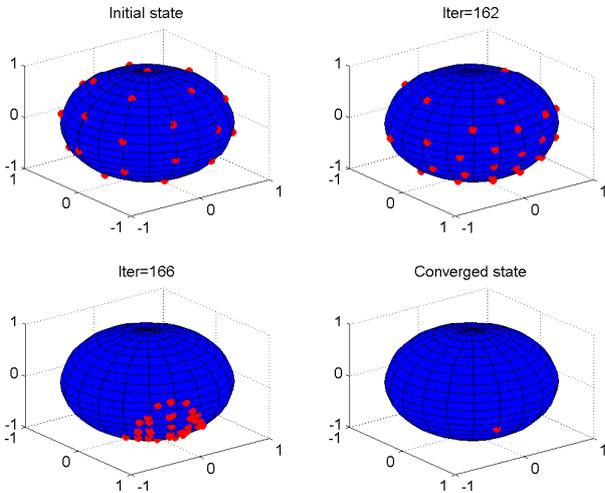


Fig. 10. Sphere Packing: Convergence to a fully aligned state with asynchronous best-response self-organization

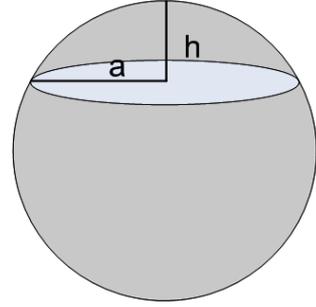


Fig. 11. A spherical cap.

sphere, move continuously, and the arcs representing the edges of the graph, span along great circles. To get from a disordered state to an ordered one, the configuration sphere has to push through one of the faces of the planar graph. The gradient descent guarantees that all nodes move towards a lowest cost point, with respect to the positions of the neighbors. At any stage of the gradient descent, consider $1 - h$ to be the fraction of the surface area of the configuration space covered by the graph, and h fraction of the area of the largest face, i.e. the part of the configuration space without any nodes. Assume that there is a cycle of L nodes surrounding this largest face. With a fixed h , the least cost configuration is one where these L nodes form a regular L -gon, and the part of the configuration space not having nodes is an approximative spherical cap. Thus an infinitesimal gradient descent would have developed to this spherical cap configuration, once it reached the point where the fraction covered by nodes is h .

Consider a spherical cap as depicted in Fig. 11. Without loss of generality, we assume that the radius of the configuration space is 1. The height of the cap is h , and the radius of the boundary circle of the cap is $a = 2\sqrt{(h - h^2)}$. The surface area of the cap is $4\pi h$, and the area of the complement cap is

$$A = 4\pi(1 - h) , \quad (15)$$

and h is indeed the fraction of the surface area occupied by the cap.

Assume that L is the length of the cycle surrounding the largest face, i.e. the cycle that the configuration sphere is squeezed through. The minimum cost regular L -gon has angle $\alpha = \pi - 2\pi/L$ between the vertices, and the length of a side is

$$s = 2a \sin \left[\frac{\pi}{L} \right] = 4\sqrt{(h - h^2)} \sin \left[\frac{\pi}{L} \right] . \quad (16)$$

This gives us directly the cost of the L -gon.

To get an approximation of the network cost (2) of disagreement, we need an estimate of the cost of the bulk of the nodes. To get an estimate of this, we assume that the communication graph is a perfect uniform sphere packing of the kind discussed above. The degree of each node is the same—each node i has $|\mathcal{N}_i| = d$ neighbors. We assume that in a minimum cost configuration it is possible that all neighbors are as far from each other. This distance scales as a square root of the area that the nodes are covering. Thus we would have

$$\|x_i - x_j\| \approx c\sqrt{1 - h}, \quad \forall (i, j) \in \mathcal{E} , \quad (17)$$

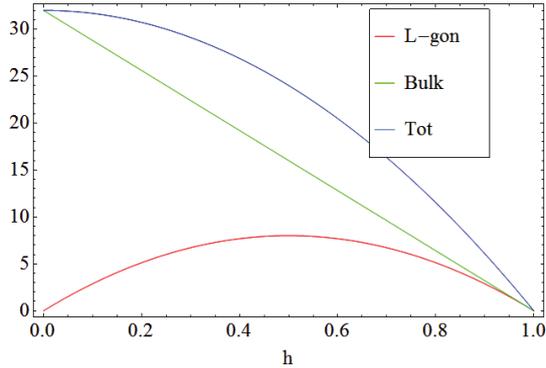


Fig. 12. Estimated minimum cost of graph covering fraction $1-h$ of sphere. Cost is power $p = 2$ of distance.

for some constant c . This gives an estimate of the cost of the nodes in the bulk. The total cost of the network configuration that is uniformly covering the fraction $1-h$ of the sphere, would thus be estimated as

$$C(h) = Ls^p + \left(L(d-2) + \frac{1}{2}(I-L)d \right) \left(c\sqrt{1-h} \right)^p. \quad (18)$$

The first term is the cost of the L -gon cycle, and the second term is the cost of the node of the bulk. Each node on the L -gon is connected to $d-2$ nodes in the bulk, and each of the $I-L$ nodes in the bulk to d neighbors. To get a feasible lower bound to the constant c , we assume the well-known Hamming bound. We assume that the area (15) covered by the nodes is divided into $I-L/2$ circles with the same area, with no space in between. The area of one such circle is $4\pi(1-h)/(I-L/2)$, and the radius is $2\sqrt{(1-h)/(I-L/2)}$. An estimate of the distance between nearest neighbors, which would make sense up to $d=6$, would then be approximately twice this radius, leading to:

$$c = \frac{4}{\sqrt{I-L/2}}. \quad (19)$$

This gives us an estimate of the cost of the minimum cost configuration covering the fraction $1-h$ of the configuration space. The estimated minimum cost for $p=2$ is depicted in Fig. 12, and the estimated minimum cost for $p=3$ in Fig. 13. It is remarkable that with $p=2$ the cost is monotonically decreasing, when the fraction of the sphere covered by the network decreases. This indicates that a configuration would align to an ordered state. However, with $p>2$, (in the example $p=3$), it is interesting that if the network is in a state with a non-trivial wrapping number, it will not be able to escape that state. Thus, it can be concluded that for a network having a planar graph topology, the ability to self-organize to an ordered state depends on the power p of the disagreement in the cost function (1). If the objective is just to find an aligned state, this power should be chosen to be $p \leq 2$.

V. CONCLUSION

We analyze the topological aspects of convergence in greedy self-organization algorithms for network synchronization and consensus problems. First, we thoroughly analyzed the case where the network is a simple cyclic graph, and the configuration space is a circle. Different identified families

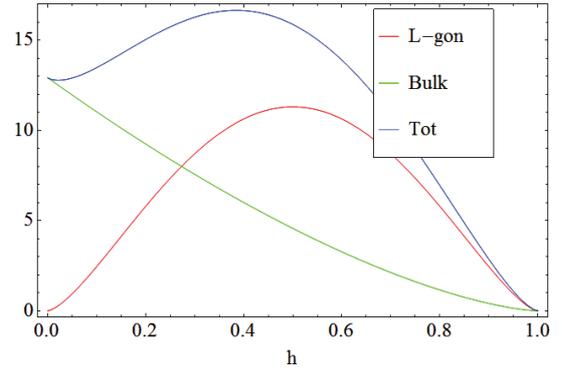


Fig. 13. Estimated minimum cost of graph covering fraction $1-h$ of sphere. Cost is power $p = 3$ of distance

of non-aligned fixed points of self-organization algorithms are identified, where the cyclic network wraps around the configuration space. Both asynchronous best response, and asynchronous gradient descent updates are considered. We interpreted these fixed points in terms of homotopy classes (“wrapping numbers”) of the mapping of the cyclic graph to the circular configuration space. The best response dynamics suffered less from such non-aligned points than the gradient descent, as the best response strategy has the possibility to change the wrapping number. However, best response dynamics are only of interest for synchronization, not for consensus.

It should be noted, though, that the presented analysis is only for cyclic graphs. For generic communication topologies, the graph typically involves multiple cycles. Topologically such a graph can be seen as a collection of cycles that are glued together. Any of these cycles may prevent synchronization due to the non-trivial topological configurations discussed here, leading to a limit-cycle behavior. In [13], for example, greedy synchronization algorithms were investigated, and limit cycle behavior was a dominating phenomenon in multi-cycle graphs. Thus, when considering synchronization of values on a circle, the topology of the circle prevents synchronization based on simple greedy algorithms. Symmetry breaking algorithms are needed for event synchronization, as is well known in the literature [11], [14]. Also, we conclude that finding consensus of a variable taking values on a ring, following the approaches of [2], [5] may sometimes be impossible due to topological obstructions.

Next, we investigated whether similar phenomena would happen in higher dimensional configuration spaces. We identified the possibility that a planar graph may wrap around an S^2 configuration space, i.e. when the configuration space is the surface of a 3D ball. However, it was found that when the cost of the distances to different neighbors are added with a power $p=2$, such configuration rapidly converge to an aligned state. Based on an analytical approximation we argued that with larger p , this would not be the case. Again, a generic graph is not planar, but can be seen as a collection of planar graphs glued together. It is likely that in a generic graph, any of the individual planar parts may prevent synchronization/consensus for $p>2$, just as any of the cycles may prevent synchronization on a circle. When the configuration spaces are spheres with higher dimension

than 2, it is likely that the higher dimensionality enables greedy algorithms to overcome the topological obstructions and converge to an ordered state. The approach of this paper may be generalized to problems with more generic compact configuration spaces [7]–[9], [16], which merits a separate study.

REFERENCES

- [1] S. Camazine, N. R. Franks, J. Sneyd, E. Bonabeau, J.-L. Deneubourg, and G. Theraula, *Self-Organization in Biological Systems*. Princeton, NJ, USA: Princeton University Press, 2001.
- [2] R. Olfati-Saber and R. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Trans. on Automatic Control*, vol. 49, no. 9, pp. 1520–1533, Sept 2004.
- [3] R. Olfati-Saber, J. Fax, and R. Murray, "Consensus and cooperation in networked multi-agent systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, Jan 2007.
- [4] P. Hovareshti, "Consensus problems and the effects of graph topology in collaborative control," Ph.D. dissertation, Univ. of Maryland, College Park, MD, USA, 2009.
- [5] R. Olfati-Saber, "Swarms on sphere: A programmable swarm with synchronous behaviors like oscillator networks," in *Proc. 45th IEEE Conference on Decision and Control (CDC'06)*, Dec 2006, pp. 5060–5066.
- [6] J. Zhu, "Synchronization of Kuramoto model in a high-dimensional linear space," *Physics Letters A*, vol. 377, pp. 2939–2943, 2013.
- [7] A. Sarlette and R. Sepulchre, "Consensus optimization on manifolds," *SIAM J. Control. Optim.*, vol. 48, no. 1, pp. 56–76, 2009.
- [8] R. Tron, B. Afsari, and R. Vidal, "Riemannian consensus for manifolds with bounded curvature," *IEEE Trans. on Automatic Control*, vol. 58, no. 4, pp. 921–934, April 2013.
- [9] R. Tron, R. Vidal, and A. Terzis, "Distributed pose averaging in camera networks via consensus on $se(3)$," in *Proc. 2nd ACM/IEEE Int. Conf. on Distributed Smart Cameras (ICDSC'08)*, Sept 2008, pp. 1–10.
- [10] R.-A. Pitaval, H.-L. Määttänen, K. Schober, O. Tirkkonen, and R. Wichman, "Beamforming codebooks for two transmit antenna systems based on optimum Grassmannian packings," *IEEE Trans. on Inf. Th.*, vol. 57, no. 10, pp. 6591–6602, Oct. 2011.
- [11] R. E. Mirollo and S. H. Strogatz, "Synchronization of pulse-coupled biological oscillators," *SIAM J. Appl. Math.*, vol. 50, no. 6, pp. 1645–1662, Dec. 1990.
- [12] A. Katok and A. B. Sossinsky, *Introduction to Modern Topology and Geometry*. Online, 2007.
- [13] J. Yu and O. Tirkkonen, "Self-organized synchronization in wireless network," in *Proc. 2nd IEEE Int. Conf. on Self-Adaptive and Self-Organizing Systems (SASO'08)*, Oct. 2008, pp. 329–338.
- [14] J. A. Acebrón, L. L. Bonilla, C. J. Pérez Vicente, F. Ritort, and R. Spigler, "The Kuramoto model: A simple paradigm for synchronization phenomena," *Rev. Mod. Phys.*, vol. 77, pp. 137–185, Apr 2005. [Online]. Available: <http://link.aps.org/doi/10.1103/RevModPhys.77.137>
- [15] N. J. A. Sloane et al., "Tables of spherical codes," published electronically at www.research.att.com/njas/packings/.
- [16] A. Sarlette, "Towards coordination algorithms on compact lie groups," Tech. Rep., 2007.