

Dynamic Self-organization and Clustering in Distributed Networked Systems for Performance Improvement

John S. Baras, Pedram Hovareshti and Senni Perumal

Abstract—We consider two closely related dynamic self-organization problems in networked control systems. Both are forms of dynamic clustering of nodes. The structure of networked control systems is often abstracted using graph theory. In this abstraction, the nodes of the graph represent the agents and the edges between them represent the relation(s) or the possibility of communication between the corresponding agents. The topology of the communication network supporting a networked control system has critical consequences for its performance. The first problem we address is the development of a distributed self-organization algorithm, resulting into a dynamic two level hierarchy of leader and regular agents, which substantially improves the convergence speed of distributed algorithms utilized by the networked control system. For the second problem, we consider the collaborative control of a group of autonomous mobile agents (e.g. vehicles, robots) supported by a mobile wireless network, consisting of many ground and a few aerial nodes. The agents collaborate to achieve a common goal or objective, like to move in a particular area and cover it, while avoiding obstacles and collisions. Building upon our earlier work on deterministic, randomized and hybrid distributed coordination algorithms we consider the communication needs of the agents, and in particular the connectivity of their communication network as they move. We develop distributed algorithms that automatically select some agents and move them appropriately so as to maintain certain degree of desired connectivity among the moving agents. We characterize the trade-off between the gain from maintaining a certain degree of connectivity vs. the combined cost of communications and the associated dynamic re-positioning of agents. We also describe classes of efficient communication topologies and in particular their similarity to dynamic small world topologies and extensions.

I. INTRODUCTION

In recent years the study of networked control systems has attracted substantial interest in the control community. Technological advancements have provided means and incentives to deploy groups of Unmanned (or Autonomous) Platforms, operating on the ground, in the air, on the water or underwater, in both military and commercial environments, in a wide variety of applications: providing coverage and connectivity to ground agents, automated highway systems, mobile sensor networks, disaster relief efforts, collaborative robotics, etc. [14]. Possible control applications to collaborative systems in many disciplines such as biology, secure computing, and sociology has been acknowledged.

J.S. Baras, P. Hovareshti and S. Perumal (baras@umd.edu, hovaresp@umd.edu, senni@umd.edu,) are with the Institute for Systems Research and the Department of Electrical and Computer Engineering, University of Maryland College Park, MD 20742, USA.

An essential aspect of many of these systems is the lack of a central control authority: distributed control rules and algorithms are often utilized due to a host of reasons including energy considerations and reliability. Autonomy is a key characteristic of such systems, which are also known as swarms. Examples of different applications of local control rules and distributed algorithms in recent literature include flocking schemes and consensus in collaborative control of groups of robots [20], [21], [22], [23], synchronization of oscillators [24], trust establishment in networks of autonomous entities [25], [26], gossip algorithms and cooperation in sensor coverage [27], [28].

In all such distributed schemes the agents are provided with simple sets of decision making algorithms or dynamics; each agent takes an action using its local information. The actions that each agent performs are also local, i.e. agents can only affect and are only affected by their neighboring nodes. The goal of the overall system is the achievement of a desired global behavior *emerging* from the local interactions. The effectiveness of these schemes depends on the following three important factors which cannot necessarily be achieved simultaneously: 1) Speed of convergence. 2) Robustness to agent/connection failures. 3) Energy/communication efficiency. It is usually desirable to achieve a favorable tradeoff level due to the particular application. An important point to notice is that the speed of convergence and robustness depend on both the structure of the network and the dynamics of the agents.

The subject of dynamic systems on graphs has also gained attention in other communities. Following the popular small world model of Watts and Strogatz [29] and the preferential attachment model of Barabasi and Albert [30], substantial research efforts have been dedicated towards constructing a scientific framework to study networks and the processes running on them. The networks under study initiate from different branches of science such as physics, computer science, biology, immunology, sociology, and chemistry and include the World Wide Web, social networks, and biological systems. Examples of such research include for instance, the studies on navigability on small world graphs, percolation on networks, and coupled dynamical oscillators. The newly emerging network science aims at methodical study of networks and processes running on them.

In this paper, we focus first on structural aspects of the network, which result in better performance of distributed algorithms. Since the decentralized nature of the algorithms

require each agent to be connected to only a few other agents, achieving high connectivity at the same time may seem counter-intuitive and even paradoxical. However, certain classes of graphs maintain this property. For example, *Watts-Strogatz small world* topologies offer a favorable tradeoff between performance (convergence speed) versus cost of collaboration (connectivity cost) [31]. Expander graphs are a class of regular sparse graphs with very good local connectivity. A certain construction class of expander graphs, Ramanujan graphs asymptotically reach the upper bound of the spectral gap.

An important property of these two classes of graph topologies is that their nodes are homogeneously distributed. Expander graphs are regular and Watts-Strogatz model of small world networks can be considered as small perturbation of certain regular matrices. What all these topologies have in common is the lack of structural bottlenecks. In many applications though the structure of the network is such that the nodes are not homogeneously distributed. There are often more “central” nodes in that they are able to communicate and influence more other nodes due to a host of problem-specific issues. For example in the case of sensor networks, the geographical distribution of the sensors and their energy supply determine which are more central. In social and economic systems the nodes’ centrality is a function of pre-existing context dependent conditions that are reinforced in the process of network formation due to the existing network’s structure and phenomena such as preferential attachment.

Hence, in many networks including those with structural bottlenecks, it is beneficial to extract community structures and provide higher level means for intercommunity communication in order to realize more efficient performance. The problem of finding communities in networks has been considered in the context of complex networks [32], and its possible relation to small world phenomenon has been pointed out [33]. Extracting efficient community structures in graphs is useful for example, in backbone construction in wireless ad hoc networks [34]. We propose a hierarchical self-organization method and show that, by utilizing it, the network is capable of running distributed algorithms with high convergence speed. Our method can also be used to improve the network performance for a given structure. We generalize the concept of social leaders to classify the agents of more importance in a network. The scheme is based on a two stage distributed algorithm, which first finds the most effective choice of local leaders, and then provides nodes with information about their location with respect to other nodes and leaders and the choice of groups to form. This leads to self-organization of the systems in the hierarchy and improves performance and response. This dynamic self-organization in a two level hierarchy is indeed a dynamic clustering scheme.

In problems regarding the control and coordination of vehicle networks, decentralized methods are preferred as centralized control requires immense communication and computational resources [19]. The collaborative control

of autonomous mobile agents can thus be viewed as a hierarchical design problem: A high level decision making and path planning module which is responsible for the communication needs of the agents and in particular maintaining the connectivity, while creating a sequence of way-points for their motion. The low level motion control computes the real control commands and actuator executions for real agents to follow the way-points generated under the dynamic and kinematic constraints.

The second and related to the first problem that we address in this paper, is the design of a high level component responsible for maintaining the communication needs of the group, and in particular the (path-) connectivity of their communication network as they move. The module has outputs to, and inputs from the higher level path planning component. This communication connectivity is crucial for instance in the typical scenario considered in [16] [17] [18], where the authors considered the scenario of a group of autonomous mobile agents (e.g. vehicles, robots) who collaborate to achieve a common goal: moving in a particular area and covering it, while avoiding obstacles and collisions. We maintain the connectivity of the group by clustering them and providing sufficient connectivity between the clusters. Building on our previous work [11], we address the problem of maintaining connectivity among ground clusters of moving agents. Among the agents in each cluster, one is designated as the *cluster-head* for the cluster. These cluster heads can be designated in a distributed manner [2], and can be equipped with multi-mode communication capabilities. In normal situations, connectivity is maintained by the cluster-heads, who send messages directed to the other cluster-heads. Therefore, the connectivity of clusters is a crucial factor in the performance of the group of agents as a whole. When direct communication between the clusters is not possible, a suggested solution is to use Aerial Platforms (APs) as relays in networks. However, the use of APs is costly and should be kept to a minimum level. Furthermore, the APs should be positioned so that the resulting network is well-connected. In other applications specific agents can move to specific locations and/or change their physical characteristics (e.g. increase their transmission power) in order to provide and maintain communication connectivity between the moving agents.

The organization of the paper is as follows. Section II provides the development of a distributed self-organization algorithm of a networked system into a two level hierarchy which allows fast convergence of various distributed algorithms. Section III considers a mobile wireless network supporting an autonomous swarm of agents and discusses a dynamic clustering algorithm used to provide the intended connectivity between the ground units using as few APs as possible. Section IV investigates the question of characterizing efficient and ‘good’ topologies, which provide better connectivity.

II. A HIERARCHICAL SELF ORGANIZING METHOD

In this section we show that a very efficient communication pattern with substantial improvement in performance is possible by a two level hierarchical scheme. The idea here is that selecting a few well connected and controlled agents which are well protected should enhance the speed of convergence of distributed schemes like consensus algorithms.

Given n agents, suppose we can divide them into K groups each having on average M members, so $n = K.M$. It is assumed that $K \leq M \ll n$. The exact sizes of K and M are problem specific and influence the performance of the algorithm. For each group suppose that we can select a ‘‘leader’’. The leaders should be able to have two properties: they should be well connected to the members of their group, and they should also be able to communicate with other leaders when necessary. If the distributed algorithm is carried out at each group separately and the leaders communicate on a higher level, the agents can enjoy faster convergence rates; the reduction of the size of each group from n to $M = O(\sqrt{N})$ results in faster intergroup convergence whereas the ease of communication between the leaders upon demand results in overall fast convergence.

We develop a distributed algorithm, which can categorize the agents as ‘‘leader’’ or ‘‘regular’’. Further, the method assigns each regular agent with an influence vector which indicates which leader has more influence on it. This provides the nodes with some global picture of the network.

Discriminating between nodes to be a leader or regular node should be done due to nodes’ centrality in the network. There is a large literature on centrality measures in networks, mostly developed in sociometry [37], [38]. Three major indices of centrality are total effect centrality, closeness centrality, and betweenness centrality which correspondingly measure the overall influence, accessibility and having control upon bottlenecks in a network. The computation of centrality indices in a network needs global information. We propose a local scheme in which nodes use their local information to come up with an approximation of their own centrality in the network and utilize this information to collaboratively decide on local leaders.

A. Distributed exploration of the graph structure

The structure of a graph plays a crucial role in properties of a distributed algorithm that is running on it. Given a graph, individual nodes have only local knowledge about its structure, which includes information about their neighboring nodes. If any node wants to either improve its own performance or a global performance measure it needs to know more about the global picture of the network. This information can be used by the node to refine its choice of neighbors in order to improve its performance.

The most complete measure of global graph structure is the adjacency matrix. Since each node has limited memory, energy, and computational capacity, they cannot store and process the adjacency matrix. Our goal is to devise a scheme to provide each node with a small vector that includes compact global information on how the node is located with

respect to the other nodes. It is desired that the scheme can be disseminated via an implementable distributed manner.

We propose a two stage algorithm for this purpose, which is carried out in a distributed manner. In the first stage nodes will collaborate to find their *social degree* [35]. This is a local measure of how ‘well connected’ each node is. Once the nodes find out their social degree, they will broadcast it locally, and they determine dynamically the K ‘‘social leaders’’ of the graph- the better connected nodes among all. The list of K social leaders is broadcast to all nodes. In the second stage, each node uses a simple iterative scheme to maintain its *influence vector*, a vector of size K which determines the influence of each social leader on it.

B. Social degrees and leader nodes

To find the leaders or the agents with the highest influence we use a generalization of a framework proposed by Blondel *et al* [35]. They define the social degree of a node as the number of the cycles of length 3 passing through that node. They also define a social leader as a node with the highest degree in its neighborhood. This can be generalized as:

Definition 1: *Social degree of order k* of a node (denoted by $SD^{(k)}(v_i)$) is defined to be the number of closed paths of length k starting from the node. For the case of $k = 2$, this reduces to the number of the neighbors of the particular node.

Definition 2: A *leader node of order k* is the node with the highest social degree of order k among its neighbors.

Definition 3: For given $0 < \alpha < 1$ and $\beta = 1 - \alpha$, a node’s *social score* is defined as $SC(v_i) = \alpha.SD^{(2)}(v_i) + \beta.SD^{(3)}(v_i)$.

Notice that each node can determine its social degree of orders 2 and 3 by a simple query from its neighbors. Since determining higher order degree requires more effort, we use the orders 2 and 3 for our present application.

In the first stage of the algorithm, each node computes its social degrees of order 2 and 3. It also queries the social degrees of its neighbors. Upon comparing its social degrees with its neighbors, if a node is found to be a leader of order 2 or 3, it broadcasts its degrees. From these data from the leader nodes, the algorithm selects K nodes $l_i, i = 1, \dots, k$ with the highest social scores, $SC(l_i)$, gives an arbitrary order to them and transmits their assigned order to them. Once a selected leader is assigned its order $1 \leq i \leq K$ it will maintain the constant vector $e_i \in R^K$. This is the unit vector with 1 in its i th entry.

Remark 2.1: The social scores as defined here, provide an approximation of the total effect centrality for each node. Formally, total effect centrality indices are defined by combining the contributions of walks of different lengths from a node using decreasing weights as the walk length increases. For example, Estrada *et al.* [39] define the centrality indices as functions of the adjacency matrix. If λ_1 is denoted as the largest eigenvalue of A , consider a real function whose

Taylor series expansion is

$$f(x) = \sum_{k=0}^{\infty} a_k x^k,$$

for $|x| < \lambda^*$, where $\lambda^* > \lambda$.

A measure of total effect centrality of a particular node i can therefore be defined as a linear combination of the number of *closed walks* of length k starting from that node (w_{ii}^k) where the corresponding weights are drawn from the Taylor series expansion of the given function f . Estrada et al. define the *functional centrality* of node i as:

$$C_f(i) = \sum_{k=0}^{\infty} a_k w_{ii}^{(k)} = (f(A))_{ii}. \quad (1)$$

It can be seen that our social scores can be considered as approximations of such Taylor series. If nodes allow for more communication and use the k^{th} order neighborhood information ($k > 2$), the resulting scores will be a more precise approximation of the global index. However, allowing for more exploration will be costly in terms of time and memory.

C. Determination of the influence vector

Our objective in this part is to associate with each of the regular nodes a vector that determines how well it is related to each of the leaders and how it is influenced by them. The amount of influence that a leader has on a local node is not only determined by their distance but also by the number of paths between them. We provide a definition for the influence vector based on the properties of random walks on graphs.

Definition 4: Consider a graph with K leaders and $n - K$ regular nodes. The influence of leader nodes l_k ($k = 1, \dots, K$) on any regular node i is the probability that a random walk that starts from i hits l_k before it hits any other leader node.

Given the leaders and the arbitrary order assigned to them, we first describe the algorithm to determine the influence vectors for each regular node. Then we will show why it converges and why it outputs valid vectors as influence vectors. We denote the influence vector of node i by $x_i \in R^K$. By $x_i^k(t)$ we mean the k th entry of the influence vector of node i evaluated at time t .

The algorithm operates as follows. The influence vector of leader l_i is first assigned to be the unit vector $x_i = e_i$. These K vectors do not vary. For all regular nodes i , x_i is initialized randomly, distributed uniformly on $[0, 1]^K$. At each iteration time $t + 1$, each regular node i , updates its influence vector entry-wise ($k = 1, 2, \dots, K$) using the following rule:

$$x_i^k(t+1) = \frac{1}{1 + n_i(t)} [x_i^k(t) + \sum_{j \in N_i(t)} x_j^k(t)] \quad (2)$$

Theorem 2.1 shows the effectiveness of the above scheme.

Theorem 2.1: If the underlying graph is connected, the iteration (2) converges to a set of unique vectors. Furthermore, $\lim_{t \rightarrow \infty} x_i^k(t)$ is equal to the probability that a random walk starting at node i hits the leader node l_k before any other leader node.

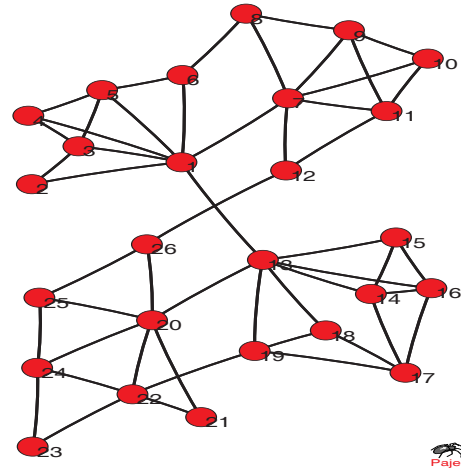


Fig. 1: A group of 26 nodes with sparse connectivity.

Proof: The proof is a direct consequence of considering the leader nodes as boundary nodes in a discrete Dirichlet problem on the graph. ■

It is worthwhile to mention that the above method has its best performance when the initial graph is nonhomogeneous and contains many structural bottlenecks. This is the case for many applications in collaborative control and sensor networks where the mission goal requires a higher concentration of nodes in certain areas. In such applications, it is beneficial for nodes in each area of concentration to self organize-around the local leader and entrust the task of inter-community communications to the local leaders.

Figure 1 is an example of a nonhomogeneous network with structural bottlenecks. The graph consists of 26 nodes. Social scores are calculated by evenly weighing social degrees of orders 2 and 3. After running the algorithm the nodes self organize into 4 groups with nodes 1, 7, 13, 20 as local leaders as shown in Figure 2. By Perron-Frobenius theory, the spectral gap of the natural random walk matrix corresponding to a graph can be considered as an indicator of the performance of distributed algorithms such as consensus algorithms on it [1]. The corresponding spectral gap for the graph of Figure 1 is 0.0535 which indicates a poor performance. However for the subgraphs in the Figure 1 the minimum spectral gap is 0.3692, much higher than that of the whole graph. Therefore in the example of consensus, running the algorithm locally, the agents can achieve local consensus much faster than the overall scheme. When the local consensus is achieved, the local leaders can negotiate among themselves with much less overhead.

Figures 3 and 4 illustrate an extreme in which there is a large difference in the resulting cluster sizes. Figure 3 is the symmetric pattern HB/can_634 from the University of Florida's sparse matrix collection [40]. It has 634 nodes and 3297 edges. Using $K = 20$, and running the algorithms results in 20 clusters as in Figure 4. The number of nodes in the clusters are not evenly distributed and the biggest cluster has as many as 179 nodes. The reason is that the nodes are

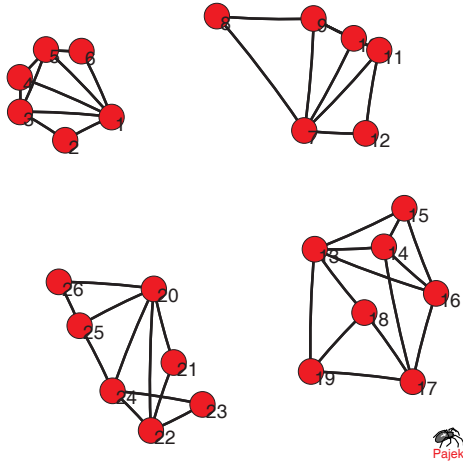


Fig. 2: Subgraphs identified by running the algorithm on the graph of Figure 1.

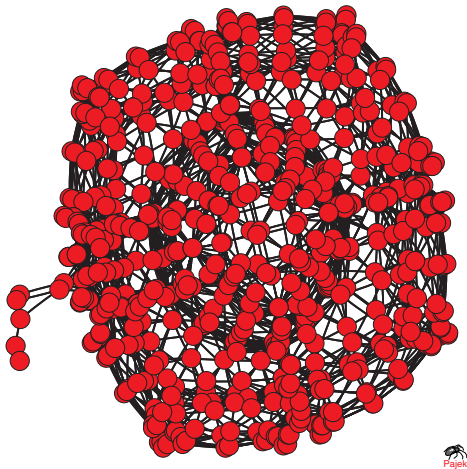


Fig. 3: A group of 634 nodes with sparse connectivity. [40]

homogeneous with cores of highly well connected nodes, which results in isolated local leaders in some parts of the network. Figures 1, 2, 3, and 4 are developed using Pajek software. [41]

III. CLUSTERING

We consider a mobile wireless network consisting of many ground nodes and a few aerial nodes, supporting the operations of an autonomous swarm of agents. Assume that all the N ground nodes have the same altitude (of 0) and form M clusters ($K_j, j = 1, \dots, M$). Aerial Platforms (APs) placed appropriately and acting as relays can be used to provide connectivity between the M disconnected ground clusters. Since APs are scarce and expensive resources, the goal is to find the minimum number of APs and their locations so that the resultant network (both between the nodes and the APs and between the APs) is connected.

The ground nodes and the APs have identical omnidirectional radios with the signal between nodes decaying as $1/R^\alpha$ where R is the distance between nodes and α is

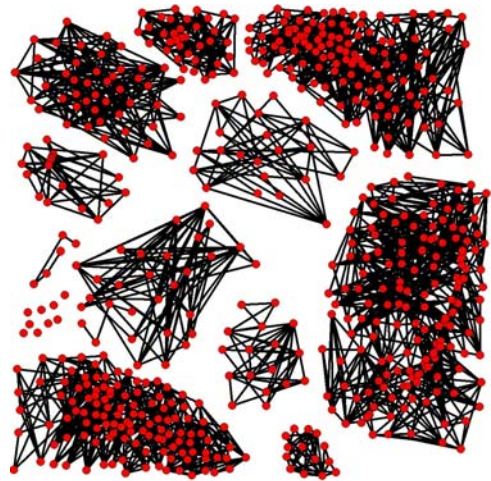


Fig. 4: Subgraphs identified by running the algorithm on the graph of Figure 3.

the path loss exponent, which depends on the environment between the nodes. The radio specifications and the path loss exponent α together determine a maximum communication distance between the nodes. α is equal to 2 (i.e., free space communication) for communication between the ground agents and APs as well as for communication between the APs. This results in a maximum communication distance of R_2 between the ground agents and APs and among the APs. Since ground nodes communicate with α strictly greater than 2 ($\alpha = 4$ for a suburban environment), the maximum communication distance R_0 between leaders is strictly less than R_2 (usually by an order of magnitude). Assume that all the APs fly at an altitude of h such that the maximum communication distance between agents and APs projected onto the ground, R_1 (given by $R_1 = \sqrt{R_2^2 - h^2}$), is greater than R_0 . Thus the problem of finding the minimum number of APs (L) and their positions can be reduced to \mathcal{R}^2 , with a_k denoting the position of the APs projected onto the ground.

A. Problem Formulation

We formulate the connectivity problem as a constrained clustering problem ([12], [13]) with a summation form distortion function ($D(K, A)$) involving the distances between the ground clusters (K) and the APs (A) and a summation form cost function ($C_1(A)$) involving only the distances between the APs (A). The resultant clustering problem is then solved using Deterministic Annealing (DA) to obtain near-optimal solutions. In order for the ground nodes and the APs to form a connected network, we need: 1) At least one node from each cluster within a radius of R_1 from an AP; and 2) Each AP is within R_2 of some other AP (i.e., the APs form a connected graph).

Assuming that the APs are numbered from 1 to L , we can make sure that they form a connected network by ensuring that any AP numbered j is connected to at least one lower numbered AP i , where $i < j$. This is used in the DA solution where when we add a new AP, we make sure that it is connected to at least one of the previously added APs. Hence

the connectivity problem can be stated as:

$$\begin{aligned} & \text{Minimize } L; \quad \text{subject to} \\ & \exists a_1, \dots, a_L; \\ & \max_{j \in \{1, \dots, M\}} \min_{\substack{g \in K_j \\ i \in \{1, \dots, L\}}} \|g - a_i\| \leq R_1 \end{aligned}$$

and,

$$\max_{l \in \{2, \dots, L\}} \min_{m < l} \|a_l - a_m\| \leq R_2$$

where $\|g - a\|$ is the l^2 -norm between points g and a on the ground. Finding the exact solution to the problem above involves an exhaustive search on the different ways in which nodes can be selected from each cluster and the ways clusters can be grouped together for coverage by a single AP all the while making sure that the APs are connected to each other. This problem is NP-hard as it is a generalization of the Euclidean *disk-cover* problem. Hence using the approximation,

$$\max(s_1, \dots, s_n) \cong (s_1^\alpha + \dots + s_n^\alpha)^{\frac{1}{\alpha}} \quad \text{for large } \alpha$$

we can convert the AP-ground node and AP-AP constraints into a summation form,

$$\begin{aligned} & \text{Minimize } L; \quad \text{subject to} \\ & \exists a_1, \dots, a_L; \quad \sum_{j=1}^M d_1(K_j, a_{u_1(j)}) \leq R_1^\alpha \\ & \text{and,} \quad \sum_{l=2}^L d_2(a_l, a_{u_2(l)}) \leq R_2^\beta \end{aligned}$$

for large α and β , where,

$$\begin{aligned} d_1(K_j, a_i) &= \min_{g \in K_j} \|g - a_i\|^\alpha \\ d_2(a_l, a_m) &= \min_{m < l} \|a_l - a_m\|^\beta \end{aligned}$$

and $u_1(j)$ is a function that assigns an AP to every cluster; $u_2(l)$ is a function that assigns the closest lower numbered AP to an AP.

Constrained clustering problems of the above form are non-convex optimization problems except in special cases. Hence the Deterministic Annealing (DA) method is used to solve the constrained clustering problem for globally near-optimal solutions. Within the framework of constrained clustering ([12], [13]), the distortion function between the ground nodes and the APs is given by $D(K, A) = \sum_{j=1}^M d_1(K_j, a_{u_1(j)})$ and the cost function among the APs is given by $C_1(A) = \sum_{l=2}^L d_2(a_l, a_{u_2(l)})$. Therefore the overall distortion function to minimize is given by:

$$\begin{aligned} D(K, A) &= \sum_{j=1}^M p(K_j) \sum_{k=1}^L p(a_k | K_j) d_1(K_j, a_{u_1(j)}) \\ &+ \lambda \sum_{l=2}^L d_2(a_l, a_{u_2(l)}) \end{aligned} \quad (3)$$

Applying the constraint on AP connectivity should increase the number of APs required for the mission. This

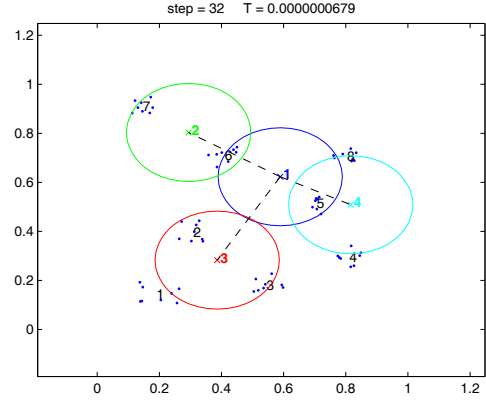


Fig. 5: Complex Scenario: AP Placement with AP-ground node connectivity and AP-AP connectivity.

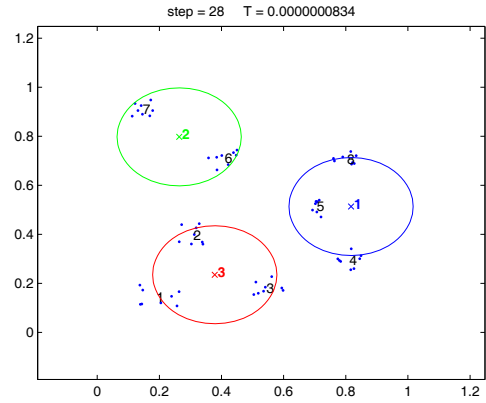


Fig. 6: Complex Scenario: AP Placement with AP-ground node connectivity but without AP-AP connectivity.

is illustrated in the following example. Consider the set up of the simulation given by figures 5 and 6. There are eight clusters each consisting of eight ground vehicles. The maximum ground to ground connectivity distance R_0 is 0.1 units, the maximum ground to AP projection distance R_1 is 0.2 units and the maximum AP to AP distance is 0.4 units. Given the parameters, all the ground vehicles within the circles around a particular AP can communicate with the AP and if the circles corresponding to APs intersect, they can communicate. Figure 5 shows that in the presence of the AP connectivity constraint a total number of 4 APs are required for full connectivity, whereas figure 6 shows that if the condition on AP connectivity is relaxed, a total number of 3 APs suffice for locally connecting clusters which are close enough.

IV. TOPOLOGY

By adding APs we have the advantage that the nodes which were previously far apart, can now communicate through APs. Prior to AP addition the neighborhood relation was based on physical proximity: any two agents which were within a distance R_0 of each other were able to

communicate and therefore would be considered neighbors. The addition of APs extends the concept of neighborhood in that two far apart agents can communicate directly through their corresponding APs. In this generalized definition of neighborhood, two agents are considered as neighbors if either they are within a distance R_0 of each other, or their corresponding APs can directly communicate. We assume that each AP is capable of short time high energy transmission upon necessity and that via a suitable medium access control and AP energy scheduling, agents which are located geographically far from each other can communicate.

This extended notion of neighborhood makes long range edges realizable. However, establishing such long range connections requires higher cost than regular short range connections. Therefore there is a trade off between the cost of maintaining long range connectivity and group performance. This is related to the efficiency of the communication topology. Small world graphs [29] are examples of such ‘efficient’ topologies. We address small world graphs next.

A. Small world graphs

The small world graphs based on the model of Watts and Strogatz [29] take a regular lattice and replace some original edges by random ones connecting nodes at ‘long distance’ with some probability $0 \leq \phi \leq 1$; i.e. by introducing ‘short cuts’. This family of graphs shows a favorable trade off between performance and cost of collaboration in a wide variety of applications, as several simulation experiments have reported.

In [7] Higham analyzed the small world phenomenon in the Watts-Strogatz model by considering the hitting time of a slightly randomly perturbed Markov chain on a ring. Building on [7], we studied consensus problems on grid-based small world graphs in [1] [2]. We showed a significant speed up in the convergence speed of consensus algorithms in Watts-Strogatz models compared to that of the grid base by perturbing the consensus weight matrix. The perturbation corresponds to considering rare transitions among non-neighboring states in the Markov chain associated with the grid. In [8], we used this probabilistic viewpoint towards understanding and quantifying small world effects on consensus convergence rates. We showed that by choosing short-cuts with low probability one can improve the convergence rate of regular grids significantly in a probabilistic sense. For other interesting works on small world graphs in the control community and consensus applications see e.g. [15], [36], [10], [5].

Here, we consider a general setting in which the base graph can be any graph. However, usually it is considered to be sparse due to obvious practical reasons having to do with notions of efficiency. We are interested in the question: is a given graph “small-worldizable”? We give a characterization of the small world effect for any general base. The following procedure gives an implicit definition of what we mean by being “small-worldizable. Denote the second largest eigenvalue modulus (SLEM) of the natural random walk matrix

F corresponding to the graph by $\mu(F)$. Other criteria for a graph to be small-worldizable are investigated in [4].

Definition 5: Small-worldizable graphs Given a connected graph \mathcal{G}_n on n vertices:

- Consider a natural random walk on this graph. Denote the corresponding Markov Chain graph as

$$F_0 = (I + D)^{-1}(A + I)$$

where A is the adjacency matrix of the graph \mathcal{G}_n and D is the diagonal matrix with each node’s degree on the corresponding diagonal.

- Perturb the zero elements of F_0 by a small ϵ and adjust its nonzero elements so that the resulting matrix F_ϵ remains stochastic.
- $G_n(V, E)$ is considered to be small-worldizable if $\frac{\Delta(F_\epsilon)}{\Delta(F_0)} \gg 1$, where $\Delta(F)$ denotes the spectral gap $1 - \mu(F)$.

Recall that if F is a primitive stochastic matrix, according to the Perron-Frobenius theorem [3], $\lambda_1 = 1$ is a simple eigenvalue with a right eigenvector $\mathbf{1}$ and a left eigenvector π such that $\mathbf{1}^T \pi = 1$, $F^\infty = \mathbf{1}\pi^T$ and if $\lambda_2, \lambda_3, \dots, \lambda_r$ are the other eigenvalues of F ordered in a way such that $\lambda_1 = 1 > \mu = |\lambda_2| \geq |\lambda_3| \geq \dots \geq |\lambda_r|$, and m_2 is the algebraic multiplicity of λ_2 , then

$$F^t = F^\infty + O(t^{m_2-1}|\lambda_2|^t) = \mathbf{1}\pi^T + O(t^{m_2-1}|\lambda_2|^t)$$

Then $\Delta = 1 - \mu(F)$ the spectral gap, so linear iterations on graphs with higher spectral gaps converge faster.

Consider a sparse connected graph \mathcal{G} , and its corresponding random walk matrix F_0 . We perturb F to get F_ϵ

$$(F_\epsilon)_{ij} = \begin{cases} \epsilon & (F_0)_{ij} = 0 \\ (1 - n\epsilon)(F_0)_{ij} + \epsilon & (F_0)_{ij} \neq 0 \end{cases}$$

where $\epsilon < 1/n$. Therefore we can write:

$$F_\epsilon = (1 - n\epsilon)F_0 + \epsilon\mathbf{1}\mathbf{1}^T.$$

Then we obtain the result:

Theorem 4.1: The graph (G) is small-worldizable if and only if $\frac{\mu}{1-\mu} \gg \frac{1}{n\epsilon}$.

Theorem 4.1 is important because by utilizing it we can decide whether a given graph is small-worldizable, what is the spectral gap gain we can get by perturbation and when (i.e for what values of parameters) the onset of the small world phenomenon occurs.

B. Expander graphs

Suppose that all APs added are within a range R_2 (as defined in section III) of each other. This means that all APs are within communication range of each other. Therefore, with proper medium access control any two nodes can send messages to any other node through one AP-AP link. In cases where lots of messages need to be transmitted between the nodes, it is no longer practical to make direct connections for each node to node transmission request.

A solution is to generate connections between the APs in such a way that the AP-level graph is an expander

graph [9]. Expander graphs have certain properties that make them suitable for our application: their large eigenvalue gap ensures fast routing and convergence of decentralized algorithms; the path diversity they provide results in robustness to link failures. We can have two methods for distributed construction of AP-level expander graphs:

- Following the approach of [9] we can form a random expander graph as a $2d$ -regular multi-graph in which the set of edges consists of d separate Hamiltonian cycles on the APs. Such a graph can be constructed distributedly and its diameter will be $O(\log_d n)$ with high probability.
- Following the approach of [6] we can form a random expander graph as the union of two spanning trees chosen independently from the uniform distribution over all spanning trees. This can be implemented simply by taking a random walk and include edges that visit previously unvisited nodes. Such a graph has a constant edge expansion with high probability.

REFERENCES

- [1] J.S. Baras and P. Hovareshti, "Effects of Topology in Networked Systems: Stochastic Models and Small worlds", *Proceedings of the 47th IEEE Conference on Decision and Control*, 2008.
- [2] J.S. Baras and P. Hovareshti. Effects of graph topology on performance of distributed algorithms for networked control and sensing. *Proceedings of the Workshop on Networked Distributed Systems for Intelligent Sensing and Control, Kalamata, Greece, (http://med.ee.nd.edu)*, 2007.
- [3] P. Bremaud. *Markov chains, Gibbs Fields, monte Carlo Simulations and Queues*. Springer Verlag, 2nd edition, 1999.
- [4] P. Duchon, N. Hanusse ,E. Lebhar, and N. Schabanel, " Could any graph be turned into a small-world?", *THEORETICAL COMPUTER SCIENCE*,355 (1): 96-103, 2006.
- [5] F. Fagnani and S. Zampieri. Randomized consensus algorithms over large scale networks. *submitted*, 2007. available at <http://calvino.polito.it/fagnani/quantcontrol/Random>
- [6] N. Goyal, L. Rademacher, S. Vempala, "Expanders via Random Spanning Trees", CORR2008.
- [7] D.J. Higham. A matrix perturbation view of the small world phenomenon. *Siam Journal on Matrix Analysis and Applications*, 25(2):429-444, 2003.
- [8] P. Hovareshti, J. S. Baras, and Vijay Gupta, "Average Consensus over Small World Networks: A Probabilistic Framework", *Proceedings of the 47th IEEE Conference on Decision and Control* , 2008.
- [9] C. Law and K. Siu, "Distributed construction of random expander networks"*IEEE Infocom Conference* 2003.
- [10] S. Patterson, B. Bamieh, and A. El Ebadi distributed average consensus with stochastic communication failures *Proceedings of 46th IEEE Conference on Decision and Control*, 4251-4220, 2007.
- [11] S. Perumal, J. S. Baras, C. J. Graff and D. G. Yee, "Aerial Platform Placement Algorithms to Satisfy Connectivity, Capacity and Survivability Constraints in Wireless Ad-hoc Networks", *To appear in the Proceedings of Milcom: 08 Assuring Mission Success*, San Diego, CA, November 17-19, 2008
- [12] K. Rose and E. Gurewitz and G. C. Fox, "Constrained Clustering as an Optimization Method", *IEEE Trans. Pattern Anal. Mach. Intell.*,15(8)785-794, 1993.
- [13] K. Rose and Miller, "Constrained clustering for data assignment problems with examples of module placement", *Proc. IEEE Int. Symp., Circuits and Systems*,pp 1937-1940 San Diego, CA, 1992.
- [14] D. A. Schoenwald, "AUVs: In space, air, water, and on the ground", *IEEE Control Sys. Mag.*, Vol. 20, No. 6 pp. 15 - -18; 2000.
- [15] A. Tahbaz-Salehi and A. Jadbabaie. Small world phenomenon, rapidly changing markov chains, and average consensus algorithm. *Proceedings of 46th IEEE Conference on Decision and Control*, 2007.
- [16] W. Xi , X. Tan and J.S.Baras, " A Stochastic Algorithm for Self-Organization of Autonomous Swarms ",*Proceedings of the 44th IEEE Conference on Decision and the European Control Conference (CDC-ECC '05)*, pp.765-770, Seville, Spain, December 12-15 , 2005.
- [17] W. Xi, X. Tan and J.S. Baras, "Gibbs Sampler-Based Self-Organization of Autonomous Swarms".*Automatica*, pp. 1107-1119, July 2006.
- [18] W. Xi and J. S. Baras, "MPC Based Motion Control of Car-like Vehicle Swarms", *Proceedings of 15th IEEE Mediterranean Conference on Control and Automation*, Athens, Greece, June 27-29, 2007.
- [19] J.S. Shamma,(Edt).*Cooperative Control of Distributed Multi-Agent Systems*, John Wiley and Sons, 2007.
- [20] V. Blondel, J. Hendrix, A.Olshevsky, and J. Tsitsiklis, "Convergence in multiagent coordination, consensus and flocking," *Proceedings 44th IEEE Conference on Decision and Control*, pp. 2996-3000, 2005.
- [21] R. Olfati-Saber, "Flocking for multi-agent dynamical systems: Algorithms and theory," *IEEE Transactions on Automatic Control*, vol. 51, no. 3, pp. 401-4203, 2006.
- [22] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Transactions on Automatic Control*, vol. 49, pp. 1520-1533, 2004.
- [23] A. Jadbabaie, J. Lin, and A. S. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Transactions on Automatic Control*, vol. 48, no. 6, pp. 988-1001, 2003.
- [24] A. Papachristodoulou and A. Jadbabaie, "Synchronization in oscillator networks: switching topologies and nonhomogeneous delays," *Proceedings of the joint 44th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC '05)*, pp. 5692-5697, 2005.
- [25] J. S. Baras and T. Jiang, "Cooperative games, phase transitions on graphs and distributed trust in manet." *Proceedings of the 43rd IEEE Conference on Decision and Control*, vol. 1, pp. 93-98.
- [26] T. Jiang and J. S. Baras, "Autonomous trust establishment," *Proceedings of 2nd International network optimization conference, Lisbon, Portugal*, 2005.
- [27] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Randomized gossip algorithms: Design, analysis and applications," *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2508-2530, 2005.
- [28] L. Xiao, S. Boyd, and S. Lall, "A scheme for robust distributed sensor fusion based on average consensus," *Proceedings of International Conference on Information Processing in Sensor Networks*, 2005.
- [29] D. Watts and S. Strogatz, "Collective dynamics of small-world networks," *Nature*, vol. 393, pp. 440-442, 1998.
- [30] A. L. Barabasi and R. Albert., "Emergence of scaling in random networks." *Science*, vol. 286, pp. 509-512, 1999.
- [31] T. Jiang and J. Baras, "Fundamental tradeoffs and constrained coalitional games in autonomic wireless," *Proceedings of 5th International Symposium on Modeling and Optimization in Mobile, Adhoc, and Wireless Networks, Limassol, Cyprus*, 2007.
- [32] M. E. J. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Phys. Rev. E*, p. 02611, 2004.
- [33] F. Chung and L. Lu, *Complex Graphs and Networks*, 2nd ed. CBMS Regional Conference Series in Mathematics; Number107 , American Mathematical Society, 2006.
- [34] S. Lee, D. Levin, V. Gopalakrishnan, and B. Bhattacharjee, "Backbone construction in selfish wireless networks," *SIGMETRICS*, pp. 121-132, 2007.
- [35] V. Blondel, C. de Kerchove, E. Huens, and P. V. Dooren., "Social leaders in graphs." *Submitted to POSTA06*, 2006.
- [36] R. Olfati-Saber, "Ultrafast consensus in small-world networks," *Proceedings of American Control Conference*, vol. 4, pp. 2371-2378, 2005.
- [37] L. C. Freeman, "Centrality in social networks: Conceptual clarification." *Social Networks*, vol. 1, pp.215-239, 1979.
- [38] N. E. Friedkin, "Theoretical foundations for centrality measures," *American Journal of Sociology*, vol. 96, No.6, pp.1478-1504, 1991.
- [39] J. A. Rodriguez-Velazquez and E. Estrada and A. Gutierrez, "Functional centrality in graphs," *Linear and Multilinear Algebra*, vol. 55, No.3, pp.293-302, 2007.
- [40] T. A. Davis, "The University of Florida Sparse Matrix Collection." Submitted to *ACM Transactions on Mathematical Software*, available at: <http://www.cise.ufl.edu/research/sparse/matrices/>.
- [41] V. Batagelj and A. Mrvar, "Pajek: Program, for Large Network Analysis", *Connections*, vol. 21, No.2, pp.47-57,1998. Project home page at: <http://vlado.fmf.uni-lj.si/pub/networks/pajek/>.