2016 IEEE Conference on Computer Aided Control System Design (CACSD)
Part of 2016 IEEE Multi-Conference on Systems and Control
September 19-22, 2016. Buenos Aires, Argentina

Follower Distribution Algorithms for Leader-Follower Networks

Begüm Şenses Cannataro*, Zhen Kan**, Warren E. Dixon***

Abstract— **In this paper, three follower distribution algorithms, (1) shortest-path follower distribution, (2) even follower distribution, and (3) hybrid follower distribution algorithms, are described for arbitrary leader-follower networks with preselected leaders. The shortest-path follower distribution algorithm is designed to minimize the path length that connects a leader to its followers while distributing followers among each leader as evenly as possible. It is shown that employing the shortest-path follower distribution algorithm with sorted greedy-based matching provides follower distributions with the smallest variance. The even follower distribution algorithm is designed to prioritize even distribution of followers among leaders over minimization of the length of path between each leader and its followers. The hybrid follower distribution algorithm is designed to prioritize goals according to a user-defined percentage, where increment of the user-defined percentage corresponds to increment in the priority of the minimization of the length of path that connects a leader to its followers over even distribution of followers. The resulting follower distributions that correspond to different user-defined percentages are illustrated on an arbitrary leader-follower network.**

I. Introduction

Multi-agent networked systems have wide potential in civilian and military applications such as transportation and distribution, logistics, surveillance, search and rescue operations, humanitarian demining, environmental monitoring, and planetary exploration [1]–[4]. Great efficiency and operational capability can be achieved in the aforementioned tasks if the workload is well distributed among agents. For instance, the allocation of sensing tasks is considered in [5] to enable efficient and periodic visit of points of interest by a team of mobile robots. Based on dynamic partitioning and distribution, large scale of robotic networks can be efficiently routed to a set of destinations in [6].

Since distributing agents to share workload can greatly improve operation efficiency and reduce implementation time and cost, various distribution algorithms have been developed. In [7] and [8], equatable partitioning policies are discussed, where the total workload is evenly distributed to each agent under the assumption that the agents and tasks are identical and any agent can be assigned to any task. In [9], a proportional workload distribution algorithm is discussed under the assumption that each agent can execute different number of tasks simultaneously. The results reported in [7]–[9], however, are only applicable to leaderless networks. While workload distribution in multi-agent networks have received great attention, workload distribution in leader-follower networks is still a problem of wide interest.

In the current work, we consider workload distribution in a leader-follower network where the leaders are superior to follower agents in the sense that leaders have exquisite access to global information. Followers are assumed to be identical and assigned to exactly one leader. In essence, follower agents can be interpreted as resources to the leader agent. To balance the workload, even distribution of followers among the leaders is desired. The path length connecting a leader and a follower varies between different pairs of followers and leaders. A longer path generally indicates larger operation cost, where communication quality generally degrades with the inter-distance of agents [10]. Therefore, the follower distribution algorithms considered in this paper have two goals: evenly distribute followers among leaders and minimize the length of each path that connects a leader to its followers. These goals, however, are difficult to simultaneously satisfy in an arbitrary leader-follower network. Therefore, three follower distribution algorithms that prioritize different goals are developed: shortest-path follower distribution, even follower distribution, and hybrid follower distribution.

The shortest-path follower distribution algorithm is designed to minimize the length of paths that connects a leader to its followers while distributing followers among each leader as evenly as possible. In other words, in the developed shortest-path follower distribution algorithm, path length between a leader and its followers is prioritized over an even distribution of followers. It is shown that employing the shortest-path follower distribution algorithm with sorted greedy-based matching provides follower distributions with the smallest variance. It is also shown that when the ratio of leader number to follower number decreases, the variance of the distribution drastically increases. The even follower distribution algorithm is designed to prioritize even distribution of followers among leaders over leader-follower path length. The comparison of follower distributions that arise from the shortest-path and the even follower distribution algorithms shows that the even follower distribution algorithm drastically decreases the variance, especially for the networks with a high ratio of leader number to follower number, while slightly increasing the length of paths that connect leaders with their followers. The hybrid follower distribution algorithm is designed to prioritize the aforementioned goals

*The Charles Stark Draper Laboratory, Cambridge, Massachusetts. Email: bcannataro@draper.com.
**Department of Mechanical and Aerospace Engineering, University of Florida, Gainesville, Florida. E-mail: kanzhen0322@ufl.edu.
***Department of Mechanical and Aerospace Engineering, University of Florida, Gainesville, Florida. E-mail: wdixon@ufl.edu.
This research is supported in part by OSD Autonomy Research Pilot Initiative project entitled A Privileged Sensing Framework and a Task Order contract with the Air Force Research Laboratory, Munitions Directorate at Eglin AFB. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the sponsoring agency.

according to a user-defined percentage, where increment of the user-defined percentage corresponds to increment of the leader-follower path length priority. The resulting follower distributions that correspond to different user-defined percentages are illustrated on an arbitrary leader-follower network.

II. Problem Formulation

In this section, leader-follower networked systems are represented by graphs. Consider a leader-follower network that consists of n followers and m leaders. In this network, nodes of the leader-follower network system are represented as $\mathcal{V} = \{v_1, \ldots, v_n, v_{n+1}, \ldots, v_{n+m}\}$ where the first n nodes correspond to the followers and the following m nodes correspond to the leaders. In addition, the interaction among agents are represented via edges, $\mathcal{E}$, where $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$. An undirected edge (v_i, v_j) exists if nodes v_i and v_j can share information with each other. The leader-follower network is described by the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{G}$ is assumed to be connected. The set of neighbors of node v_i is defined as $\mathcal{N}_i = \{v_j | v_j \in \mathcal{V}, (v_i, v_j) \in \mathcal{E}\}$ and the degree of v_i is $|\mathcal{N}_i|$ where $|\mathcal{N}_i|$ is the cardinality of $\mathcal{N}_i$. The set of agents is denoted by $\mathcal{S}_k$ where k indicates that the length of the shortest path that connects any agent in $\mathcal{S}_k$ with at least one leader is k. Set $\mathcal{S}_0$ contains the leaders of the network and $\mathcal{S}_1$ contains the followers that are directly connected to at least one leader. If k is greater than 1, the followers in $\mathcal{S}_k$ are indirectly connected to at least one leader through a path of length k. Although some followers may be connected to multiple leaders with different number of edges, each follower belongs to exactly one $\mathcal{S}_k$ where k is the smallest possible value for each follower.

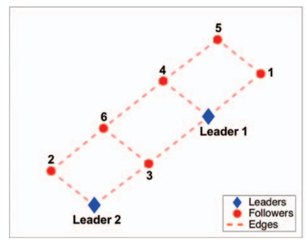
III. Follower Distribution Algorithms

The goals of the follower distribution algorithms are to evenly distribute followers among leaders and to minimize the length of each path that connects a leader to its followers. In this Section, three algorithms are discussed in detail: (1) the shortest-path follower distribution, (2) the even follower distribution, and (3) hybrid follower distribution algorithms.

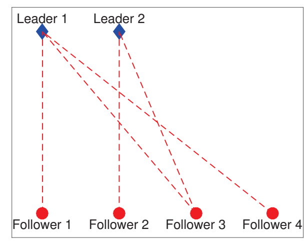
A. *Shortest-Path Follower Distribution Algorithm*

The shortest-path follower distribution (SPD) algorithm is designed to minimize the number of edges in each path that connects a leader to its followers while distributing followers among each leader as evenly as possible. In other words, the SPD algorithm minimizes connecting edges between leaders and its followers with priority over an even distribution of followers. As a result of this priority, even if a follower is connected to multiple leaders, it can only be assigned to one of the leaders with the shortest connecting path. To obtain leader-follower pairs with the shortest connecting path, agents in the network are divided into levels, $\mathcal{S}$. As explained in Section II, $\mathcal{S}_k$ is a set of agents where k indicates the shortest path that connects any agent in $\mathcal{S}_k$ to at least one of the leaders. After the network is divided into levels, each agent in $\mathcal{S}_{k-1}$ is matched with a directly connected agent in $\mathcal{S}_k$, where k is an integer between 1 and the graph diameter, via a greedy-based matching. In greedy-based matching, the matching is completed in order of the agent node numbers. Therefore, if an agent in $\mathcal{S}_{k-1}$ is directly connected to multiple agents in $\mathcal{S}_k$, it is matched with the agent that has the smallest node number.

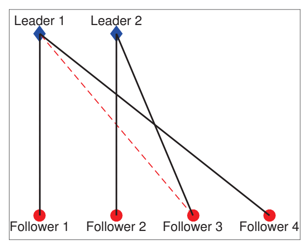
The complexity of SPD algorithm with greedy-based matching is between $O(na)$ and $O(na^2)$ where na is the total number of agents. Although the length of paths between a leader and its followers is minimized with a computationally inexpensive method, the followers may not be distributed among leaders as evenly as possible. In Fig. 1 the uneven distribution of followers via greedy-based matching is illustrated on a network with 2 leaders, 6 followers, and 10 edges. In this network, S_0 contains Leader 1 and Leader 2, S_1 contains Followers 1-4, and S_2 contains Followers 5 and 6. Fig. 1(b) gives the bipartite graph that illustrates the edges between the S_0 and S_1 agents. As shown in Fig. 1(c), the followers in S_1 can be evenly distributed among leaders if Follower 1 and 4 are assigned to Leader 1, while Follower 2 and 3 are assigned to Leader 2. As shown in Fig. 1(d), the SPD algorithm with greedy-based matching, however, assigns 3 followers to Leader 1 and 1 follower to Leader 2.



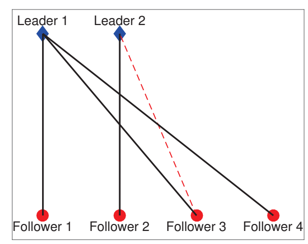
(a) Multi-agent network



(b) Bipartite graph of S_0 and S_1



(c) Even Follower Distribution



(d) Follower distribution via greedy-based matching

Fig. 1. Illustration of uneven distribution of followers via greedy-based matching

As observed in Fig. 1, in SPD algorithm with greedy-based matching the followers are not distributed among leaders as evenly as possible. To provide an even follower distribution, the greedy-based matching between $\mathcal{S}_{k-1}$ and $\mathcal{S}_k$ can be performed after sorting agents in $\mathcal{S}_{k-1}$ in an ascending order according to the degree of the leader nodes (if $k = 1$) or the modified degree of follower nodes (if $k > 1$). While the degree of a leader node, v_l, is simply the number of followers in its adjacency list, $|N_l|$, the modified degree of a follower node, v_i, is calculated as

$$md_i = |N_i| + f(l_i) \tag{1}$$

where md_i is the modified degree of v_i, N_i is the set that contains the neighbors of v_i, l_i is the leader of v_i, and $f(l_i)$ is the total number of followers that are assigned to l_i. As mentioned before, when k is equal to 1, S_{k-1} consists of leaders. Therefore, only the degree of leaders affects the sorting. On the other hand, when k is greater than 1, S_{k-1} contains followers that are already distributed among leaders. Therefore, both the degree of Follower i and the number of followers that are assigned to the leader of Follower i affect the sorting. The greedy-based matching with sorting is employed to the network given in Fig. 1(a) and it is observed that the matching between S_0 and S_1 results in even distribution of followers as given in Fig. 1(c).

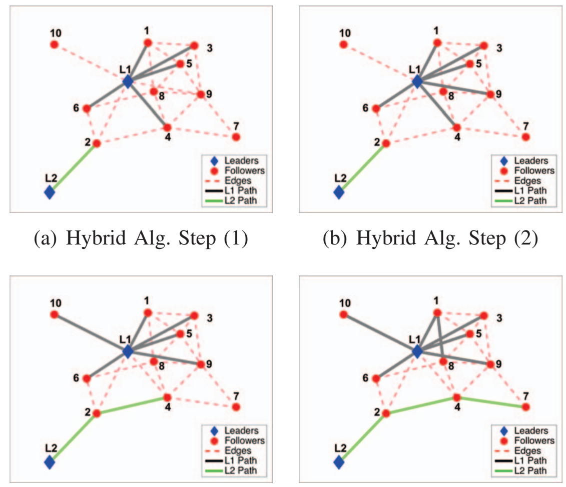
In SPD algorithm, the followers are distributed among leaders level-by-level. Therefore, if the number of followers that are connected to each leader is uneven in a level, the aforementioned sorting cannot prevent uneven distribution of followers. In Section III-B, a follower distribution algorithm that guarantees an even follower distribution for any arbitrary leader-follower network is discussed.

B. Even Follower Distribution Algorithm

In this section, an even follower distribution (EFD) algorithm is designed to prioritize an even distribution of followers among leaders over minimization of the number of connecting edges between each leader and its followers. Employing the EFD algorithm requires a current level set, $\mathcal{C}$, and a future level set, $\mathcal{F}$. At the beginning of EFD algorithm, the current level set of Leader i, $\mathcal{C}_i$, contains only the leader itself, while $\mathcal{F}$ of every leader is empty. After obtaining an initial $\mathcal{C}$ and $\mathcal{F}$ for every leader, the follower-leader matching process of the EFD algorithm begins. In each matching iteration, each leader is matched with a follower that is directly connected to at least one agent in the corresponding $\mathcal{C}$. The matching is completed in order of the agent node numbers. Therefore, if an agent in $\mathcal{C}$ is directly connected to multiple agents in the network, it is matched with the agent that has the smallest node number. The follower that is matched with Leader i is added to the future level set of Leader i, $\mathcal{F}_i$. If all of the candidate followers for Leader i (the followers that are directly connected to at least one agent in $\mathcal{C}_i$) are already assigned to another leader, $\mathcal{C}_i$ and $\mathcal{F}_i$ are both updated where the updated $\mathcal{C}_i$ is equal to the previous $\mathcal{F}_i$ and the updated $\mathcal{F}_i$ is empty. After the update, the EFD algorithm continues with the aforementioned matching iteration until all of the followers are assigned to a leader. In the EFD algorithm, the even distribution of followers is guaranteed since exactly one follower is assigned to each leader at every iteration with the exception of iterations where at least one $\mathcal{C}_i$ is empty. In Section IV, it is shown that the exceptional iterations with at least one empty $\mathcal{C}_i$ only slightly increases the variance.

C. Hybrid Follower Distribution Algorithm

In this section, a hybrid follower distribution (HFD) algorithm is discussed. As mentioned before, the goals of the follower distribution algorithms are to evenly distribute followers among leaders and to minimize the length of each path that connects a leader to its followers. The HFD algorithm is designed to prioritize these goals according to a user-defined percentage, where an increment of the user-defined percentage corresponds to an increment in the priority of the minimization of the connecting edges between leaders and followers over an even distribution of followers. The steps of HFD algorithm are explained and illustrated on the network given in Fig. 2. The steps of HFD algorithm are



(a) Hybrid Alg. Step (1) (b) Hybrid Alg. Step (2)

(c) Hybrid Alg. Steps (3)-(5) (d) Final follower distribution

Fig. 2. Follower distribution among leaders during the steps of hybrid follower distribution algorithm.

given as follows.

(1) Distribute followers by using shortest-path follower distribution algorithm with sorted greedy-based matching (see Section III-A) until

$$\frac{max(f) - min(f)}{n_f} \geq p \tag{2}$$

where f is a vector that gives the number of followers that are assigned to each leader, n_f is the total number of followers in the network, and p is the aforementioned user-defined percentage. In Fig. 2(a), the resulting follower distribution of the example network after employing Step 1 of the HFD algorithm with user-defined percentage 0.4 is illustrated. As seen in Fig. 2(a), $f = [5\ 1]$ (5 followers are assigned to Leader 1 and 1 follower is assigned to Leader 2), $\max(f) = 5$, $\min(f) = 1$. Therefore, Eq. 2 is satisfied and Step 1 is terminated.

(2) Keep distributing followers by using the even follower distribution algorithm (see Section III-B) until
   (a) all of the followers are assigned to a leader or
   (b) all of the candidate followers of Leader i in the current level are assigned to another leader and even after updating the current level, an available candidate follower cannot be found.

If Step 2 is terminated after Option (a) is satisfied, terminate the HFD algorithm. Otherwise go to Step 3. In Fig. 2(b), the resulting follower distribution of the example network after employing Step 2 is illustrated. As seen in Fig. 2(b), Step 2 is terminated since all of the candidate followers of the second leader are assigned to the first leader.

(3) First, find the neighbors of agents that are in $\mathcal{F}_i$, $\mathcal{NF}_i$. Second, find the leaders that correspond to the neighbors, $\mathcal{LF}_i$. Fig. 2(b) shows that $\mathcal{F}_2 = 2$, $\mathcal{NF}_2 =$ [4 6], and $\mathcal{LF}_2 = 1$.

(4) Search an alternative matching for the leaders in $\mathcal{LF}_i$. The alternative matching for the Leader $j \in \mathcal{LF}_i$ is an unmatched follower that is connected to at least one agent in $\mathcal{C}_j$. If an alternative match is found, go to Step 5, otherwise go to Step 6. In the example network given in Fig. 2, Step 4 searches for an alternative matching for Leader 1. As seen in Fig. 2(b), Follower 10 is directly connected to the first leader, and it is not assigned to any leader. Therefore, Follower 10 is an alternative matching for Leader 1.

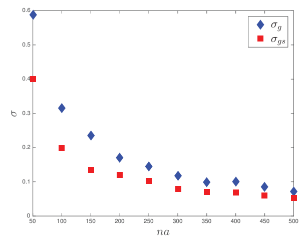
(5) Match Leader i (the same Leader i mentioned in Step 2 - Option b) with a follower that was previously assigned to Leader j (the same Leader j mentioned in Step 4) and return to Step 2. The comparison of Figs. 2(b) and 2(c) shows that at the end of Step 5, Follower 4 is no longer assigned to Leader 1, now it is assigned to Leader 2. Instead of Follower 4, Follower 10 (the alternative matching) is assigned to Leader 1.

(6) Find the leader with the maximum number of followers in $\mathcal{LF}_i$, assign one of its followers to Leader i, and return to Step 2. During follower distribution of the network shown in Fig. 2, the HFD algorithm never enters this step.

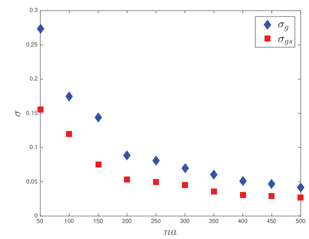
IV. Simulation Results

In this section, simulation results are provided to illustrate the performance of the follower distribution algorithms explained in Section III. The performance of the follower distribution algorithms are evaluated by comparing the normalized variance of the follower distribution among leaders, σ, where the variance is normalized by the number of followers in each network. A zero normalized variance corresponds to evenly distributed followers, while a large normalized variance corresponds to a distribution that is far away from the possible even distribution.

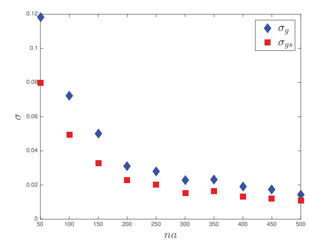
The results of SPD algorithm are obtained by using randomly generated networks with various number of total agents (summation of total number of leaders and followers), na, number of edges, ne, and number of leaders, nl. In these random networks, na changes between 10 and 500, ne/na changes between 4/3 and 3, and nl/na changes between 1/100 and 1/3. For every na, ne, and nl combination, the results are computed as the average of the results that arise from 10 randomly generated networks with the same na, ne, and nl combination. The normalized variances that arise from the SPD algorithm with sorted and unsorted greedy-based matching are denoted as σ_{gs}, and σ_g, respectively. The results of the SPD algorithm have similar patterns for every ne/na and nl/na value. Therefore, the general pattern of the results is explained by showing plots that correspond to two values of ne/na and nl/na.



(a) σ_g, σ_{gs} vs. na ($ne/na = 2$, $nl/na = 1/8$)



(b) σ_g, σ_{gs} vs. na ($ne/na = 3$, $nl/na = 1/8$)

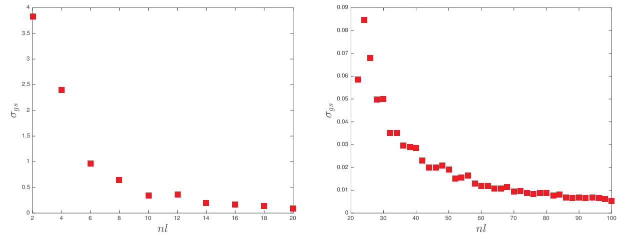


(c) σ_g, σ_{gs} vs. na ($ne/na = 2$, $nl/na = 1/3$)

Fig. 3. The normalized variance, σ, that arises from shortest-path follower distribution algorithm with sorted greedy-based matching, σ_{gs} and unsorted greedy-based matching, σ_g vs. total number of agents, na, for various nl/na and ne/na values where nl gives the number of leaders and ne gives the number of edges.

Fig. 3 gives σ_g and σ_{gs} as a function of na for various values of ne/na and nl/na values. Fig. 3 shows the following two important results: employing sorting increases the performance of SPD algorithm with greedy-based matching and σ that arises from SPD algorithm decreases when na increases. Comparing Figs. 3(a) and 3(b) shows that σ decreases when ne/na increases and comparison of Figs. 3(a) and 3(c) shows that σ decreases when nl/na increases. In both cases, the decrease in σ is due to the fact that increment in ne/na and nl/na increase the number of leaders that a follower can be assigned to.

To explain the effect of the number of leaders on σ in more detail, another simulation is executed on randomly generated networks with $na = 250$ and $ne/na = 2$. As in the earlier simulation, σ is calculated by taking the average over results that arise from 10 randomly generated networks with the same nl. Fig. 4 shows σ_{gs} vs. nl where $na = 250$ and $ne/na = 2$. As mentioned before, the follower distributions with the smallest σ are obtained by performing SPD with sorted greedy-based matching. Therefore, in this simulation only the results that arise from sorted greedy-based matching are illustrated. While Fig. 4(a) gives results that correspond to networks with less than or equal to 20 leaders, Fig. 4(b)



(a) σ_{gs} vs. nl, $2 \leq nl \leq 20$ ($na = 250$, $ne/na = 2$)
(b) σ_{gs} vs. nl, $22 \leq nl \leq 100$ ($na = 250$, $ne/na = 2$)

Fig. 4. The normalized variances that arise from greedy-based follower selection algorithm with sorting, σ_{gs} vs. nl for $ne/na = 2$ and $na = 250$ where ne is the number of edges and na is the total number of agents.

gives results that correspond to networks with greater than 20 leaders. As seen in Fig. 4, when nl decreases, σ increases. When nl is less than 10, σ drastically increases. The reason for this drastic increase is explained on a randomly generated network with 2 leaders, 10 followers, and 24 edges ($ne/na = 2$) given in Fig. 5. As seen in Fig. 5, while 9 followers are

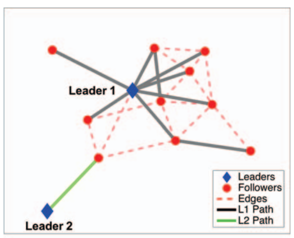


Fig. 5. Follower distribution via shortest-path follower distribution algorithm with sorted greedy-based matching on a randomly generated network with 2 leaders and 10 followers where the dashed lines show the edges between agents, L1 Path shows the paths between the first leader and its followers and L2 Path shows the paths between the second leader and its followers.

assigned to the first leader, only 1 follower is assigned to the second leader. In this example, the uneven distribution of followers is due to fact that in the SPD algorithm the level of followers has priority over an even distribution of followers. As a result of this priority, even if a follower is connected to multiple leaders, it can only be assigned to one of the leaders with the shortest connecting path. As observed in Fig. 5, the second leader has only one directly connected follower and all of its level 2 followers are directly connected to the first leader. Therefore, all of its level 2 followers are assigned to Leader 1. As shown in the example network (see Fig. 5) when nl decreases σ may increase drastically since leaders may have an uneven number of followers with the shortest connecting path.

In the network given in Fig 5, even follower distribution among leaders can be obtained by performing the EFD algorithm. The follower distribution that arises from the EFD algorithm is illustrated in Fig. 6. As shown in Fig. 6, the EFD algorithm assigns the same number of followers for both leaders but a trade-off exists between even distribution of followers and the number of edges in each path that connects a leader to its followers. The comparison of Fig. 5 and 6 shows that the follower distribution that arises from the SPD algorithm with sorted greedy-based matching contains 8 Level 1 followers and only two Level 2 followers while the follower distribution that arises from the EFD algorithm contains 6 Level 1 followers, 2 Level 2 followers, and 2 Level 3 followers.

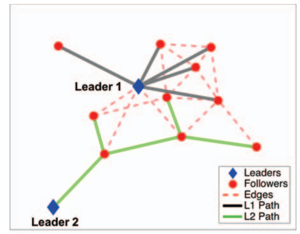


Fig. 6. Follower distribution via even follower distribution algorithm on a randomly generated network with 2 leaders and 10 followers where the dashed lines show the edges between agents, L1 Path shows the paths between the first leader and its followers and L2 Path shows the paths between the second leader and its followers (The same network that is given in Fig. 5).

To understand the trade-off between even distribution of followers and the number of edges in each path that connects a leader to its followers, the EFD algorithm is employed on the same randomly generated networks that are used to obtain Fig. 4. Fig. 4 shows the normalized variance of the follower distribution among leaders that arises from the EFD algorithm, σ_e vs. nl where $na = 250$ and $ne/na = 2$. Moreover, Fig. 8 shows the comparison of follower distributions that arise from the SPD algorithm with sorted greedy-based matching and the EFD algorithm in terms of the normalized number of followers, nfl, where the number of followers in each level is normalized by the total number of followers in each network. The results given in Figs. 7 and 8 show

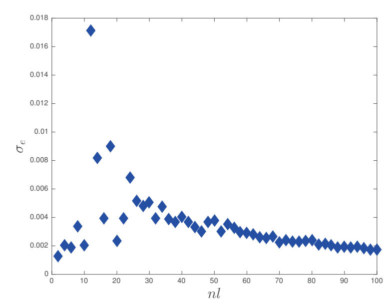


Fig. 7. The normalized variances that arise from even follower distribution algorithm, σ_e, vs. nl for $ne/na = 2$ and $na = 250$ where ne is the number of edges and na is the total number of agents.

that the EFD algorithm drastically decreases the variance especially for the networks with high ratio of leader number to follower number while slightly increasing the total number of edges that connects leaders with their followers.

The HFD algorithm allows the user to decide the priorities in the resulting follower distribution with a percentage input. As mentioned before the increment in the percentage corresponds to increment in the priority of the minimization of the length of path that connects a leader to its followers over even distribution of followers. Fig. 9 illustrates the effect of

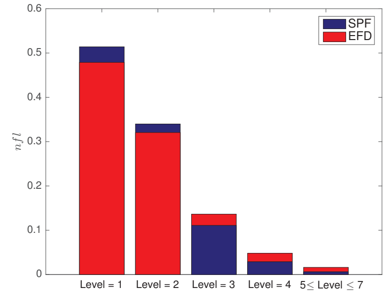
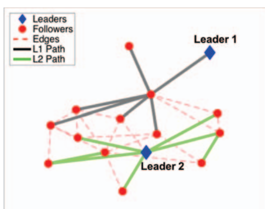
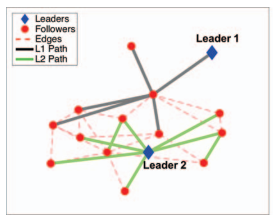


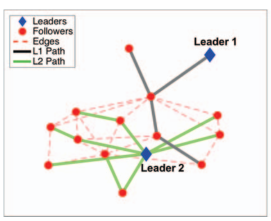
Fig. 8. Comparison of follower distributions that arise from shortest-path follower distribution algorithm with sorted greedy-based matching, SPD, and even follower distribution algorithm, EDF, in terms of the normalized number of followers, nfl, where the number of followers in each level is normalized by the total number of followers in each network.



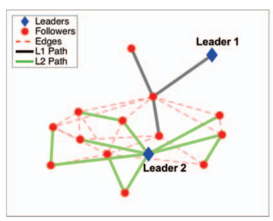
(a) $p = 0$



(b) $p = 0.2$



(c) $p = 0.4$



(d) $p \geq 0.6$

Fig. 9. Follower distribution via hybrid follower distribution algorithm with various user-defined percentages, p, on a randomly generated network with 2 leaders and 13 followers where the dashed lines show the edges between agents, L1 Path shows the paths between the first leader and its followers and L2 Path shows the paths between the second leader and its followers.

the user-defined percentage, p, in the HFD algorithm on a randomly generated network with 2 leaders, 13 followers, and 30 edges. For the network given in Fig. 9, the vector of number of followers that are assigned to each leader, f, is given for various values of p as

$$p = 0, \quad f = [6, 7] \qquad p = 0.2, \quad f = [5, 8]$$
$$p = 0.4, \quad f = [4, 9] \qquad p \geq 0.6, \quad f = [3, 10].$$

As seen from f vectors that correspond to different p values, when p increases, the variance of the follower distribution increases. For the network given in Fig. 9, the vector of number of followers in each level, L, is given for various values of p as

$$p = 0, \quad L = [6, 7] \qquad p = 0.2, \quad L = [7, 6]$$
$$p = 0.4, \quad L = [7, 5, 1] \qquad p \geq 0.6, \quad L = [7, 6].$$

As seen from L vectors that correspond to different p values, when p goes from 0 to 0.2, the total length of paths that connect leaders and their followers decreases. The follower distribution algorithm that arises from the HFD algorithm when p is equal to 0.2, has the minimum total length of paths that connect leaders with their followers. Therefore, increasing p after 0.2 does not decrease the total length of paths.

V. Conclusion

Strategies to distribute followers among pre-selected leaders in an arbitrary leader-follower network was considered. Three follower distribution algorithms that prioritize evenly distribution of followers among leaders or minimization of the number of edges in each path that connects a leader to its followers are developed. Simulation results were provided to demonstrate the performance of the developed follower distribution algorithms. Results showed that employing the SPD algorithm with sorted greedy-based matching provides follower distributions with the smallest variance. In a comparison of follower distributions that arise from the SPD and EFD algorithms showed that the EFD algorithm drastically decreases the variance, especially for the networks with high ratio of leader number to follower number, while slightly increasing the length of paths that connect leaders with their followers. The resulting follower distributions that correspond to different user-defined percentages using the HFD algorithm were illustrated on an arbitrary leader-follower network.

References

[1] M. Mesbahi and M. Egerstedt, *Graph theoretic methods in multiagent networks*. Princeton University Press, 2010.

[2] W. Ren, R. W. Beard, and E. M. Atkins, "Information consensus in multivehicle cooperative control," *IEEE Control Syst. Mag.*, vol. 27, pp. 71–82, Apr. 2007.

[3] Z. Kan, L. Navaravong, J. Shea, E. Pasiliao, and W. E. Dixon, "Graph matching based formation reconfiguration of networked agents with connectivity maintenance," *IEEE Trans. Control Netw. Syst.*, vol. 2, no. 1, pp. 24–35, Mar. 2015.

[4] J. R. Klotz, Z. Kan, J. M. Shea, E. L. Pasiliao, and W. E. Dixon, "Asymptotic synchronization of a leader-follower network of uncertain Euler-Lagrange systems," *IEEE Trans. Control Netw. Syst.*, vol. 2, no. 2, pp. 174–182, Jun. 2015.

[5] X. Zheng and S. Koenig, "Reaction functions for task allocation to cooperative agents," in *Int. Conf. Auton. Agents Multiagent syst.* International Foundation for Autonomous Agents and Multiagent Systems, 2008, pp. 559–566.

[6] L. Liu and D. A. Shell, "Large-scale multi-robot task allocation via dynamic partitioning and distribution," *Auton. Robots*, vol. 33, no. 3, pp. 291–307, 2012.

[7] M. Pavone, E. Frazzoli, and F. Bullo, "Distributed policies for equitable partitioning: Theory and applications," in *IEEE Conf. Decis. Control*. IEEE, 2008, pp. 4191–4197.

[8] M. Pavone, A. Arsie, E. Frazzoli, and F. Bullo, "Distributed algorithms for environment partitioning in mobile robotic networks," *IEEE Trans. Autom. Control*, vol. 56, no. 8, pp. 1834–1848, 2011.

[9] H. Sayyaadi and M. Moarref, "A distributed algorithm for proportional task allocation in networks of mobile agents," *IEEE Trans. Autom. Control*, vol. 56, no. 2, pp. 405–410, 2011.

[10] A. Gonzalez-Ruiz, A. Ghaffarkhah, and Y. Mostofi, "A comprehensive overview and characterization of wireless channels for networked robotic and control systems," *Journal of Robotics*, vol. 2011, 2012.