A Dynamic Flocking Algorithm with a Restrictive Partnership Model to Support Mobile Ad Hoc Networks

Abdullah Konak, *Member, IEEE* and Sadan Kulturel-Konak

Abstract— **This paper presents an improved flocking algorithm to increase the connectivity of a mobile ad hoc network using autonomous and intelligent agents. Flocking algorithms usually aim to simulate realistic movements of a group of agents. In this paper, however, agents use a flocking algorithm to find a solution to a computationally very difficult optimization problem in real-time as the topology of the network changes due to the mobility of users. In the improved flocking algorithm, agents select their interaction partners based on the Gabriel Graph and adjust their flocking behavior parameters dynamically. A simulation study is conducted to compare the performance of the improved flocking algorithm with a previous algorithm. Computational studies show that the recommended strategies are quite effective.**

I. Introduction

An ad hoc network is a communication network that is established spontaneously by a set of devices that can communicate without requiring a fixed communication infrastructure. As wireless communication technologies are increasingly integrated into a variety of devices, mobile ad hoc networks (MANETs) have found a wide range of viable real-life applications such as military missions, emergency response, and search/rescue maneuvers [1-5].

The topology of a MANET is dynamic because its nodes move freely. New nodes may join the network, existing nodes may disappear, or wireless communication links vanish when nodes move out of the range. Therefore, maintaining an acceptable level of Quality of Service (QoS) in MANETs is a challenging task. The foremost challenge is to ensure that a MANET provides connectivity to all of its users at all times. In the literature, several papers propose the use of special mobile nodes to augment the topology of a MANET dynamically as the nodes move [6-11]. These special nodes, called *agents*, monitor the state of the network and dynamically adjust their locations to support the connectivity of other nodes (or user nodes). For this system to work, agent nodes should have information regarding the whereabouts of the other nodes and make periodic decisions regarding where to move. Furthermore, the information exchange and location optimization should be performed in real-time and depend on the limited computing capability of nodes.

To guide the deployment decisions of agents, two distinct types of methods are proposed in the literature: centralized and distributed. In the centralized methods [12-14], the network is assumed to have a central management system that is aware of the locations of all nodes and capable of communicating with the agents at all times. The central management system optimizes the positions of the agents and directs them where to move periodically. Although it is possible to identify the locations of nodes and communicate with agents in some MANET applications, the resulting mathematical problem is still very to difficult solve. Therefore, metaheuristics, such as Particle Swarm Optimization, are frequently used as the optimization engine in the centralized methods.

In the distributed methods [8, 9], agents are only aware of the nodes that they can directly communicate with. Agents make their deployment decisions independently based on interactions with their local neighbors. They use simple flocking rules, including modified versions of the cohesion and separation rules defined by Reynolds [15] in the original flocking algorithm, to determine their new locations as the topology of the network changes. In the literature, flocking algorithms are frequently applied to the cooperative control of mobile robots [16-23]. In the context of mobile robots, the function of flocking algorithms is to keep the robots together and avoid obstacles while performing a task. However, the problem under consideration in this paper is quite different because user nodes are assumed to move randomly.

This paper extends the flocking algorithm of Konak et al. [9] in two ways: (*i*) the interaction partners of an agent is selected based on the Gabriel Graph and (*ii*) each agent adjusts the parameters of its own flocking algorithm dynamically based on the crowdedness of its neighborhood. These modifications aim to address some of the problems observed in the previous algorithms [8, 9] such as clustering of agents and unnecessary links established by agents.

II. Problem Description

Consider a MANET $G(t)$ with a node set $N(t)$ and edge set $E(t)$ at time t (i.e., $G(t)=(N(t),E(t))$. There are two types of nodes, user nodes (set $U(t)$) and agent nodes (set $A(t)$). User nodes are assumed to move randomly; and therefore, the topology of the network is dynamic and random. The nodes communicate over wireless links that are established if two nodes are within one another's communication range. Let point $\mathbf{p}_i(t)=(x_i(t),y_i(t))$, $x_i(t)\in\mathbb{R}$ and $y_i(t)\in\mathbb{R}$, represent the location of node i at time t. Then, edge set $E(t)$ of the network at time t is defined as

$$E(t)=\{(i,j): i,j\in N(t), i\neq j, d_{ij}(t)\leq \min(R_i,R_j)\}$$

where R_i is the communication range of node i, and $d_{ij}(t)$ is the Euclidean distance between nodes i and j (i.e., $d_{ij}(t)=\|\mathbf{p}_i(t)-\mathbf{p}_j(t)\|$ where $\|\mathbf{p}\|$ denotes the Euclidean norm of vector $\mathbf{p}$.

The mission time of the network is divided into T discrete time intervals. The objective of agent nodes is to maximize the connectivity of user nodes as the network topology changes during the mission time. Let binary variable τ_{ijt}=1 if there is a path between user nodes i and j at time t, and τ_{ijt}=0 otherwise. The mathematical formulation of the problem is expressed as follows:

$$\text{Max} \quad Q = \frac{1}{T}\sum_{t=1}^{T} \frac{200 \sum_{i\in U(t)} \sum_{j\in U(t):i<j} \tau_{ijt}}{(|U(t)|(|U(t)|-1)}$$

subject to:

$$\|\mathbf{p}_i(t)-\mathbf{p}_i(t-1)\| \le V_i \quad \forall i \in A(t), t=1,\ldots,T$$

The objective function of the problem represents the average percent of connected (directly or indirectly) user node pairs throughout the mission time. The constraint of the problem states that each agent i can move a maximum of V_i unit distance between two consecutive time periods.

The mathematical formulation given above is helpful to express the problem, but it is difficult to solve optimally. Konak et al. [10] present a nonlinear mixed-integer programming formulation to determine the optimal locations of agent nodes at a period t given their locations at period t-1 and solve the formulation for each period independently throughout the mission time. However, they report that this approach can only be applied to small-sized networks with a limited number of agents. Therefore, a metaheuristic approach based on PSO is suggested [7, 10]. Recently, Magán-Carrión et al. [24] have developed and tested a PSO algorithm with a mobility prediction to locate relay nodes dynamically in real-life robot swarms. Magán-Carrión et al. [25] also introduce a multi-stage approach for locating relay nodes on a MANET to maximize reachability and the network throughput.

III. PROPOSED FLOCKING ALGORITHM

In the flocking algorithm defined by Konak et al. [9], the movement of an agent node is defined as

$$\mathbf{p}_i(t+1)=\mathbf{p}_i(t)+\frac{\min(V_i,\|\mathbf{v}_i(t)\|)}{\|\mathbf{v}_i(t)\|}\mathbf{v}_i(t) \qquad (1)$$

where $\mathbf{v}_i(t)=(vx_i(t),vy_i(t))$ is the velocity vector of agent i, indicating the direction and the distance that agent i intends to travel from period t to t+1. The velocity vector $\mathbf{v}_i(t)$ is calculated based on three flocking behaviors, separation, cohesion, and exploration which are shaped by the agent's interactions with its neighbors at time t. In [9], the set of interaction partners of agent i includes all nodes that the agent has a direct link with (i.e., all topological neighbors of agent i, which is denoted as $N_i(t)$). Therefore, an agent may be heavily influenced by their interaction partners that are clustered together. Such clustered neighbors may lead the agent to overreact. In this paper, we define a new interaction partnership model based on the Gabriel Graph. In this new partnership model, agent i interacts with node j if the following two conditions hold:

- $d_{ij}(t) \le \min(R_i, R_j)$
- There is no other node within the circle where the line segment connecting $\mathbf{p}_i(t)$ and $\mathbf{p}_j(t)$ is a diameter.

For agent i, let $NG_i(t)$ denote the set of nodes that satisfy the above two conditions. Fig. 1 illustrates an example of determining interaction partners of an agent (Node 1) based on the Gabriel Graph. Assuming that all nodes are within the range of one another, $N_1(t)$ would include all nodes in the figure. However, $NG_1(t)$ includes only nodes 2, 3, 7, and 8, indicated by the red colored edges in the figure. Node 6 is not considered as an interaction partner of Agent 1 because node 3 resides within the circle between agent 1 and node 6. Similarly, nodes 4 and 5 are not interaction partners of Agent 1 due to the location of node 3.

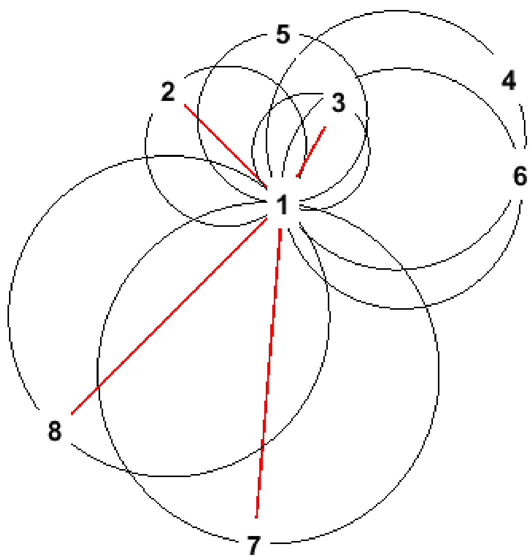

Figure 1. Determination of interaction partners of Node 1 based on the Gabriel Graph.

In the flocking algorithm, agents exhibit different behaviors while they interact with their user and agent interaction partners. Therefore, it is necessary to identify the user and agent neighbors of agent i. Let $U_i(t)$ and $A_i(t)$ represent the sets of user and agent topological neighbors of agent i at time t (i.e., $N_i(t) = U_i(t) \cup A_i(t)$). In [9], the interactions of agent i with its neighbors are adjusted by four static parameters: separation weight, cohesion weight, user target distance, and agent target distance. It is suggested that these parameters are tuned based on the density of the network, communication ranges, and mobility of nodes. For example, if the network is dense, it is recommended that the separation behavior has a higher weight than the cohesion behavior in the movement of an agent. These flocking parameters are identical for all agents and static during the entire course of the mission time.

In the proposed flocking algorithm defined in this paper, the cohesion and separation behaviors are self-tuned by two dynamic parameters $SU_i(t)$ and $SA_i(t)$ that represent the target distances that agent i seeks to maintain between its user and agent interaction partners, respectively. The velocity of agent i with respect to its interaction partner node j is described as follows:

$$\mathbf{v}_{ij}(t)=\begin{cases}0.5\dfrac{(d_{ij}(t)-SA_i(t))}{d_{ij}(t)}(\mathbf{p}_j(t)-\mathbf{p}_i(t)) & j\in A_i(t)\\ \dfrac{(d_{ij}(t)-SU_i(t))}{d_{ij}(t)}(\mathbf{p}_j(t)-\mathbf{p}_i(t)) & j\in U_i(t)\end{cases} \tag{2}$$

If the distance between agent i and node j is more than the target distance, then $\mathbf{v}_{ij}(t)$ suggests agent i to move closer to node j (i.e., the cohesion behavior). Otherwise, agent i exhibits the separation behavior and moves away from node j. The total velocity of agent i due to all of its interaction partners are given as

$$\mathbf{v}_i(t)=\begin{cases}\alpha\sum_{j\in NG_i(t)}\mathbf{v}_{ij}(t)+(1-\alpha)\mathbf{v}_i(t-1) & NG_i(t)\neq\{\}\\ \alpha V_i(\cos(\theta),\sin(\theta))+(1-\alpha)\mathbf{v}_i(t-1) & NG_i(t)=\{\}\end{cases} \tag{3}$$

where α is a memory parameter (between 0 and 1), and it indicates how much of the previous information gained by an agent is incorporated into the current velocity decision. If an agent has no interaction partners, then the agent moves with its maximum velocity in the direction of angle θ, which is selected randomly and uniformly between 0 and 2π.

Target distance parameters $SU_i(t)$ and $SA_i(t)$ in (2) are updated within their upper and lower bounds based on the crowdedness of $N_i(t)$ as follows:

- If $|A_i(t)| \leq 2$ Then $SA_i(t)=SA_i$
- If $|U_i(t)| \leq 2$ Then $SU_i(t)=SU_i$
- If $|A_i(t)| \geq 3$ Then $SA_i(t)=\min(1.1\times SA_i(t), R_i)$
- If $|U_i(t)| \geq 5$ Then $SU_i(t)=\min(1.1\times SU_i(t), R_i)$

where SA_i and SU_i are the lower bound of $SA_i(t)$ and $SU_i(t)$, respectively. The lower bounds can be set based on the communication range, the terrain, and the density of the network. Note that although $NG_i(t)$ is used in (3) to compute the velocity of agent i, the size of its distance-based neighborhood determines how $SA_i(t)$ and $SU_i(t)$ are updated. It is also important to note that $NG_i(t)$ and $N_i(t)$ can be very different depending on how the neighboors of agent i located around the agent. The target distances of an agent are increased if it resides in a crowded region where the agent is less likely to support the connectivity of the network. Thereby, the agent is prompted to move away from crowded areas.

Another advantage of the dynamic update of the target distance parameters is that no additional flocking parameters are needed to prioritize between the cohesion and separation behaviors since agents dynamically update their behaviors based on the neighborhood size of the agent. The procedure for agent movements is given as follows:

```
MoveAgent(i, t) {
   Determine N_i(t), A_i(t), and U_i(t) at p_i(t)
   Identify interaction partners, i.e., NG_i(t), in N_i(t)
   Calculate velocity v_i(t) using (3)
   Calculate new location p_i(t+1) using (1)
   Move agent i to new location p_i(t+1)
}
```

IV. Computational Experiments

The simulation environment defined in [9] is used to test the performance of the enhancements to the flocking algorithm [9]. The user nodes are randomly moved within a circle of the radius of 300-unit distance. In each time step t, user node $i\in U(t)$ changes its direction angle $\theta_i(t)$ randomly between 0 and 2π with a probability of ρ. Then, the user node travels a random distance between $V_{\min}$ and $V_{\max}$ in the direction of angle $\theta_i(t)$. If the new location of the user node is outside the simulation area, a new angle $\theta_i(t)$ is randomly generated until the node stays within the simulation area. The procedure for simulating user movements is given below.

```
MoveUser(i, t) {
   Repeat {
      If Rand(0,1)≤ρ Then θ_i(t)=Rand(0, 2π)
                    Else θ_i(t)=θ_i(t-1)
      p_i(t+1)=p_i(t)+Rand(V_min,V_max)(Cos(θ_i(t)), Sin(θ_i(t)))
   } Until (p_i(t+1) is in the simulation area)
}
```

In simulation experiments, the four versions of the flocking algorithm were compared as given in Table 1. Algorithm F is the original flocking algorithm defined in [9], which is used as the benchmark in this paper. Algorithm FG is a version of Algorithm F where interaction partners are selected based on the Gabriel Graph. Both Algorithms FD and FDG use the dynamic target distance update procedure introduced in this paper, and interaction partners are chosen based on the Gabriel Graph in Algorithm FDG.

TABLE I. Versions of the Flocking Algorithms Tested

Interaction Partners	Target Distance Update	Algorithm
$N_i(t)$	Fixed	F
$NG_i(t)$	Fixed	FG
$N_i(t)$	Dynamic	FD
$NG_i(t)$	Dynamic	FDG

In experiments, various size test networks were used with 20, 30, 40, and 50 user nodes and 2, 4, 6, 8, and 10 agents. The parameters of the user mobility simulation were $V_{\min}$=5, $V_{\max}$=10, and ρ=0.1, and R_i=100 for all nodes in all test networks. The flocking parameters of Algorithms FD and FDG were α=0.90, V_i=10, SA_i=75, and SU_i=50. For benchmark algorithm F as well as FG, the cohesion and separation weights were set to 1 while the exploration weight was set to zero (i.e., agents did not have any exploration behavior if they are connected to other nodes). Algorithms F and FG used static target distances of SA_i=75 and SU_i=50 and α=0.90 for each agent *i*.

The simulation was run for *T*=1000 with a warm-up period of 50 for 100 random replications for each test network. In each random replication, the four algorithms were tested against the same movement patterns of user nodes.

Table 2 presents the results of the simulation experiments. The average percent of connected user pairs during the mission time (i.e., the objective function value (*Q*) of the problem given in Section 2) was calculated for each random simulation replication. In Table 2, the mean results of 100 replications are provided for only Algorithm F. In addition, the percent difference in the mean values of Algorithm F and the others are provided for a quick identification of improvements in the objective function.

TABLE II. THE RESULTS OF THE SIMULATION STUDY

(|U|,|A|)	Mean Q(F)	(FG-F)/(F)	(FD-F)/(F)	(FGD-F)/(F)
(20,2)	23.73	2.4%	2.3%	2.0%
(20,4)	28.89	6.9%	6.5%	7.2%
(20,6)	34.39	13.4%	10.3%	13.2%
(20,8)	40.21	19.9%	14.4%	20.5%
(20,10)	46.33	26.3%	17.4%	27.1%
Average		**13.8%**	**10.2%**	**14.0%**
(30,2)	36.35	3.5%	4.7%	4.1%
(30,4)	42.65	8.9%	10.9%	9.6%
(30,6)	49.46	14.7%	16.4%	16.6%
(30,8)	56.14	20.2%	21.1%	23.1%
(30,10)	62.38	25.5%	25.9%	28.3%
Average		**14.6%**	**15.8%**	**16.4%**
(40,2)	55.12	4.0%	6.5%	5.9%
(40,4)	62.21	8.4%	12.2%	11.5%
(40,6)	68.85	12.4%	17.1%	16.8%
(40,8)	74.57	16.5%	21.4%	20.7%
(40,10)	79.72	19.4%	23.5%	23.8%
Average		**12.1%**	**16.1%**	**15.7%**
(50,2)	75.34	2.8%	5.2%	4.7%
(50,4)	80.76	5.5%	8.9%	8.3%
(50,6)	85.02	8.0%	11.6%	10.8%
(50,8)	88.73	10.0%	13.2%	13.1%
(50,10)	91.17	11.5%	14.4%	13.9%
Average		**7.6%**	**10.7%**	**10.2%**

The results in Table 2 show that the methods proposed in this paper to improve the performance of a flocking algorithm for guiding deployment decisions of agents that aim to increase the connectivity of a MANET are effective. Both the dynamic target distance update procedure and the interaction partnership method based on the Gabriel Graph, used either individually in algorithms FG and FD or together in Algorithm FDG, increased the average connectivity of the test networks significantly.

In all test cases, percent improvement was observed with the increasing number of agents. One of the problems of Algorithm F is that multiple agents may be attracted to the same clusters of densely connected user nodes and not able to break out from them. Hence, these agents will be less likely to contribute to the overall connectivity of the network. Selecting interaction partners of agents based on the Gabriel Graph encourages agents to collaborate implicitly. For example, if two agents are in close proximity and have many mutual topological neighbors, the rules of the Gabriel Graph will prevent these agents to share many identical interaction partners. Thereby, the two agents can focus on supporting a different set of users. The dynamic target distance approach also prompts agents to move away from densely connected segments of the network.

V. CONCLUSION

The computational experiments show that the flocking algorithm performs better if some of the clustered neighbors of agents are ignored in determining their movements. In this paper, the interaction partners of an agent are selected based on the Gabriel Graph seems to establish sufficient but not excessive interconnections for agents to select their partners. Thereby, an agent movement is not heavily influenced by a group of its topological neighbors that are clustered tightly. In addition, some flocking parameters such as the number of interaction partners and determining which partners to select are not required. Therefore, the proposed flocking algorithm has a limited number of parameters to tune.

The computational experiments support that the flocking algorithm with dynamic target distances increases the connectivity of networks significantly. The proposed self-tuning ability of agent behaviors not only requires fewer parameters but also prevents agents to be attracted to densely connected user clusters. The proposed flocking algorithm is quite straightforward and can be implemented in a decentralized and asynchronous manner.

Future research may include designing and testing more comprehensive agent behaviors and alternative ways of identifying interaction partners.

REFERENCES

[1] N. Aschenbruck, M. Frank, P. Martini, and J. Tolle, "Human Mobility in Manet Disaster Area Simulation - a Realistic Approach," in *Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks (LCN'04)*, 2004, pp. 668-675.

[2] I. Chlamtac, M. Conti, and J. J. N. Liu, "Mobile Ad Hoc Networking: Imperatives and Challenges," *Ad Hoc Networks,* vol. 1, pp. 13-64, 2003.

[3] C. R. Dow, P. J. Lin, S. C. Chen, J. H. Lin, and S. F. Hwang, "A Study of Recent Research Trends and Experimental Guidelines in Mobile Ad-Hoc Network," in *Proceedings of the 19th International Conference on Advanced Information Networking and Applications (AINA'05)*, 2005, pp. 72-77.

[4] M. Forodigh, P. Johansson, and P. Larsson, "Wireless Ad Hoc Networking-the Art of Networking without a Network," *Ericsson Review,* vol. 4, pp. 248-263, 2000.

[5] D. Simplot-Ryl and I. Stojmenovic, "Data Communications and Topology Control in Wireless Ad Hoc Networks," *Ad Hoc Networks,* vol. 3, pp. 507-508, 2005.

[6] R. R. Choudhury, K. Paul, and S. Bandyopadhyay, "An Agent-Based Connection Management Protocol for Ad Hoc Wireless Networks," *Journal of Network and Systems Management,* vol. 10, pp. 483-504, 2002.

[7] O. Dengiz, A. Konak, and A. E. Smith, "Connectivity Management in Mobile Ad Hoc Networks Using Particle Swarm Optimization," *Ad Hoc Networks,* vol. 9, pp. 1312-1326, 2011.

[8] A. Konak, "Improving Network Connectivity in Emergency Ad Hoc Wireless Networks," in *The 11th International Conference on Information Systems for Crisis Response and Management (ISCRAM 2014)*, The Pennsylvania State University, University Park, PA, 2014, pp. 36-44.

[9] A. Konak, G. Buchert, and J. Juro, "A Flocking-Based Approach to Maintain Connectivity in Mobile Wireless Ad Hoc Networks," *Applied Soft Computing,* vol. 13, pp. 1284–1291, 2013.

[10] A. Konak, O. Dengiz, and A. E. Smith, "Improving Network Connectivity in Ad Hoc Network Using Particle Swarm Optimization and Agents," in *Wireless Network Design: Optimization Models and Solution Procedures (International Series in Operations Research and Management Sciences)*. vol. 158, J. Kennington, E. Olinick, and D. Rajan, Eds., ed: Springer 2011, pp. 247-267.

[11] C. H. Ou, K. F. Ssu, and H. C. Jiau, "Connecting Network Partitions with Location-Assisted Forwarding Nodes in Mobile Ad Hoc Environments," in *10th Pacific Rim International Symposium on Dependable Computing (PRDC'04)*, Papeete, Tahiti, French Polynesia, 2004, pp. 239–247.

[12] K. Chadrashekar, M. R. Dekhordi, and J. S. Baras, "Providing Full Connectivity in Large Ad-Hoc Networks by Dynamic Placement of Aerial Platforms," University of Maryland TR 2004-21, 2004.

[13] S. Hauert, L. Winkler, J.-C. Zufferey, and D. Floreano, "Ant-Based Swarming with Positionless Micro Air Vehicles for Communication Relay," *Swarm Intelligence,* vol. 2, pp. 167-188, 2008.

[14] H. Zhu, A. L. Swindlehurst, and K. J. R. Liu, "Smart Deployment/Movement of Unmanned Air Vehicle to Improve Connectivity in Manet," in *IEEE Wireless Communications and Networking Conference*, Vegas, NV, USA, 2006, pp. 252-257.

[15] C. W. Reynolds, "Flocks, Herds and Schools: A Distributed Behavioural Model," in *Proceedings of the Conference held at Computer Graphics 87*, Pinner, UK, 1987, pp. 71-87.

[16] X. W. Zhao, B. Hu, Z. H. Guan, C. Y. Chen, M. Chi, and X. H. Zhang, "Multi-Flocking of Networked Non-Holonomic Mobile Robots with Proximity Graphs," *IET Control Theory & Applications,* vol. 10, pp. 2093-2099, 2016.

[17] D. Sakai, H. Fukushima, and F. Matsuno, "Flocking for Multirobots without Distinguishing Robots and Obstacles," *IEEE Transactions on Control Systems Technology,* vol. 25, pp. 1019-1027, 2017.

[18] T. Nguyen, T. T. Han, and H. M. La, "Distributed Flocking Control of Mobile Robots by Bounded Feedback," in *54th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, 2016, pp. 563-568.

[19] F. J. Mendiburu, M. R. A. Morais, and A. M. N. Lima, "Behavior Coordination in Multi-Robot Systems," in *IEEE International Conference on Automatica (ICA-ACCA)*, 2016, pp. 1-7.

[20] W. Wang, C. Wen, and J. Huang, "Adaptive Flocking Control of Multiple Nonholonomic Mobile Robots with Limited Communication Ranges," in *9th Asian Control Conference (ASCC)*, 2013, pp. 1-6.

[21] P. Dasgupta, C. Ke, and F. Li, "Flocking-Based Distributed Terrain Coverage with Dynamically-Formed Teams of Mobile Mini-Robots," in *IEEE Swarm Intelligence Symposium*, 2009, pp. 96-103.

[22] N. Xiong, Y. Li, L. T. Yang, J. H. Park, Y. Yang, and S. Tao, "Fast and Efficient Formation Flocking for a Group of Autonomous Mobile Robots," in *IEEE International Symposium on Parallel and Distributed Processing*, 2008, pp. 1-8.

[23] C. Yang Quan, W. Zhongmin, and L. Jinsong, "Automatic Dynamic Flocking in Mobile Actuator Sensor Networks by Central Voronoi Tessellations," in *IEEE International Conference Mechatronics and Automation, 2005*, pp. 1630-1635.

[24] R. Magán-Carrión, J. Camacho, P. García-Teodoro, E. F. Flushing, and G. A. Di Caro, "Drns: Dynamical Relay Node Placement Solution," in *Advances in Practical Applications of Scalable Multi-Agent Systems. The Paams Collection: 14th International Conference, Paams 2016, Sevilla, Spain,* Y. Demazeau, T. Ito, J. Bajo, and M. J. Escalona, Eds., ed Cham: Springer International Publishing, 2016, pp. 273-276.

[25] R. Magán-Carrión, R. A. Rodríguez-Gómez, J. Camacho, and P. García-Teodoro, "Optimal Relay Placement in Multi-Hop Wireless Networks," *Ad Hoc Networks,* vol. 46, pp. 23-36, 2016.