The Thirty-First International Florida
Artificial Intelligence Research Society Conference (FLAIRS-31)

Informal Team Assignment in a Pursuit-Evasion Game

David W. King, Jason M. Bindewald, Gilbert L. Peterson*
david.king@afit.edu, jason.bindewald@afit.edu, gilbert.peterson@afit.edu
Air Force Institute of Technology
Wright-Patterson AFB, OH 45433

Abstract

Control architectures and algorithms for large autonomous swarms are receiving increased research interest. Control of swarm systems becomes more difficult as the size of the agent swarm increases, making centralized control approaches inadequate. This paper presents the informal team assignment algorithm. By leveraging agent roles and signaling actions, the algorithm provides a local agent mechanism leading to the emergence of cooperative teams. Tested in a modified pursuit-evasion domain, simulation results demonstrate that agent roles and inter-agent signaling spontaneously create small collaborative agent teams dedicated to shared task accomplishment. The result is in higher win ratios for signal and role capable swarms.

1 Introduction

As countries such as China and the United States (US) look into integrating large swarms of autonomous agents into their military forces (Kania 2016), developing algorithms and architectures for controlling agent swarms increase in importance for safety and defense. When the size of an agent swarm increases, centralized controllers quickly become overwhelmed by the rising complexity of agent task assignments, communication, and interactions. In order to meet this growing challenge, this paper presents the informal team assignment algorithm (ITAA). The algorithm leverages agent roles and inter-agent signals to provide decentralized control of an autonomous swarm engaged in a global task.

Tested in a modified two-dimensional (2D) pursuit evasion scenario, simulations revealed that agent signaling led to the formation of small teams of pursuing agents, invoking the temporary leadership behavior noted in pigeon flocks (Chen, Liu, and Zhang 2016). This is a notable finding, as the algorithm created collaborative teams without predefined leader-follower roles or a centralized control authority.

The rest of the paper is organized as follows. Section 2 reviews current autonomous swarm control research. Section 3 formally presents the informal team assignment algorithm. Section 4 describes the experimental variables and scenarios, with Section 5 providing data analysis and discussion. Section 6 concludes the work.

*Distribution Statement A - Cleared for Public Release.

2 Background

Finding methods for decentralized control of autonomous swarms is an active area of research (McCune et al. 2013; Cao et al. 2014; Tolba, Ammar, and Rajasekaran 2016). Tan, et al. (Tan and Zheng 2013) proposed that decentralized control provided through individual agent behaviors produces a scalable and robust complex system. This work views agent behaviors, defined as *roles*, as the central component for system scalability. Furthermore, sharing information through signaling enables a swam to alter its structure through collaborative team formation, making the swarm robust to dynamic changes in the environment.

Tolba, et al. (Tolba, Ammar, and Rajasekaran 2016) used a combination of flocking behaviors and limited signaling to control a swarm of autonomous underwater vehicles (AUVs) engaged in search operations. Each agent executed an independent search pattern while using flocking rules (Reynolds 1987) to avoid other agents and environmental obstacles. The agents used signaling only once they located an object to attract other vehicles to the site. All of the agents possessed the same role, *search*, with predefined behaviors. Although a homogeneous swarm may be useful for single tasks, such as locating an object, a complex code rewrite may be necessary to add additional tasks or behaviors. Additionally, in their current set-up, the first agent to locate an object would attract all available agents to its location, making multi-object search, and by extension multi-goal achievement, difficult.

Duan, et al. (Duan, Li, and Yu 2015) modeled the dynamics of unmanned combat aerial vehicles (UCAV) using a predator-prey particle swarm optimization (PSO) technique. They tested two autonomous teams against one another, where one was set on destroying the other. Although the agents were allowed to dynamically select enemy targets, the UCAVs did not communicate their selected actions. Additionally, both teams consisted of the same type of agents with the attacking force given a slight speed advantage. The authors successfully modeled the given scenario; however, dynamic reassignment of roles and communication between agents was ignored.

Beard, et al. (Beard et al. 2002), proposed an all encompassing solution for Unmanned Aerial Vehicle (UAV) cooperative control. The UAVs were tasked with moving between multiple target locations in a dynamic threat environment with the primary goal of having a team of UAVs arrive simultaneously on the edge of each target's radar detection area. In a manner similar to (Tolba, Ammar, and Rajasekaran 2016), each UAV possessed one role, *target engagement*, and, although UAV's coordinated their actions, the authors accomplished it through a centralized communications controller. However, the complexity of coordinating a large autonomous swarm quickly overwhelms a central control mechanism making such an approach for large swarms infeasible.

Alexopolous, et al. (Alexopoulos, Schmidt, and Badreddin 2015) appear to be the first to use the pursuit-evasion domain to study the influence of agent roles on the performance of autonomous agent teams. Their agents possessed two roles, *herd* and *pursue*. Agents determined their current role after solving an n-player discrete-time deterministic game selecting the role which led to a victory for their team. The selection of roles, where one agent could help corral the evading agent by *herding* them, yielded positive results, indicating that agent access to multiple roles improves agent swarm performance in dynamic environments.

The ITAA, presented in the following section, incorporates the idea of role assignment introduced by (Alexopoulos, Schmidt, and Badreddin 2015), et al., while including agent signaling found in (Tolba, Ammar, and Rajasekaran 2016) and (Beard et al. 2002). Agent roles and signals serve as a mechanism for decentralized control. As agents possess multiple roles, multiple sub-swarms (i.e. teams) and multi-task accomplishment become realizable. Signals allow agents to share information, affecting each agent's role selection and team membership. Populations of teams may shift autonomously as agents receive, and digest, new signals. Decentralized control makes the agent swarm scalable, while roles and signals make them robust.

3 Algorithm

Emergence research (Holland 2012) provided the inspiration for the ITAA. Emergence theorizes that the aggregated interactions between simple agents give rise to complex behaviors (Bedau and Humphreys 2008; Holland 1998; Mitchell 2006; Serugendo et al. 2004). Holland (Holland 2012) proposes that agents use signals, both direct and indirect, to interact and affect one another and, by extension, the system as a whole. Bonabeau, et al. (Bonabeau, Dorigo, and Theraulaz 1999) believe that the ability of agents to change roles, in response to changes in the environment, makes natural systems (e.g. ant and bee colonies) successful. The ITAA incorporates both ideas.

Algorithm 1 provides pseudocode for the ITAA. Before running the algorithm, one assigns each agent a set of roles, $R = \{r_1, r_2, \ldots, r_n\}$. Each role contains a threshold setting, $T = \{t_1, t_2, \ldots, t_n\}$, which is composed of coded logic that determines when an agent changes its role. Finally, one assigns a set of actions, $A = \{a_1, a_2, \ldots, a_j\}$, to each role.

Each agent processes the ITAA at each time step. First, the agent processes sensory input, this includes incoming signals from other agents. The agent stores this data and iterates through its potential roles (line 1). If a role meets a predefined threshold (line 2), such as detection of an object, the agent changes its role accordingly (line 3). Once an agent finishes role selection, it executes the actions assigned to the selected role (line 6).

Algorithm 1 Informal Team Assignment Algorithm (ITAA)

Input: Sensory Data
Output: None

```
1: for all r in Roles do
2:     if threshold t met for r then
3:         currentRole = RoleAssignment()
4:     end if
5: end for
6: Execute actions assigned to currentRole
```

With the ITAA, agents make decisions autonomously, free from centralized control. Agents invoke a Markov assumption for role selection as they rely upon current data without consideration for past events. Thresholds represent the probability that the agent moves to another role (i.e. state). Actions, such as signal propagation, influence other agent decisions. The interplay between roles and signals creates informal agent teams that grow and shrink in response to environmental stimuli.

4 Experiments

A modified pursuit-evasion scenario was created to test the scalability and robustness of an agent swarm using the ITAA. The scenario consisted of an autonomous agent swarm (*pursuing* agents) tasked with defending a defined coordinate (i.e. nest) from an autonomous invading agent (*evading* agent). This scenario is similar to pursuit-evasion games (PEGs), where a team of pursuing agents tries to capture an evading agent (Alexopoulos, Schmidt, and Badreddin 2015). However, this work differs in two ways from other UAV based PEG scenarios (Alexopoulos, Schmidt, and Badreddin 2015; Li 2006; Vial et al. 2001) in that (1) large pursuit teams (up to 20 agents) were tested and (2) evading agents possessed superior speed.

Scenario:

Figure 1 shows the initial set up for each simulation. Initial positions for both evading and pursuing agents were randomly assigned within their respective territories. This kept evading and pursuing agents from spawning too close to one another. Random placement also dispersed agents across the grid enabling multiple angles of entry for the evading agent, overcoming any possible pattern formations that could be advantageous to either side. Evading agents sought to reach the nest center, while pursuing agents needed to detect and intercept the evading agent to prevent it from reaching the nest. Evading agents won if they reached the nest. Pursuing agents won if they caught the evading agent.

Agent Implementation:

Algorithm 2, Nest Defense, presents the ITAA implementation for pursuing agents in a nest defense scenario. Evading agents always moved towards the nest center. If they detected a pursuing agent, they calculated a vector that moved them away from the pursuing agent but toward the nest

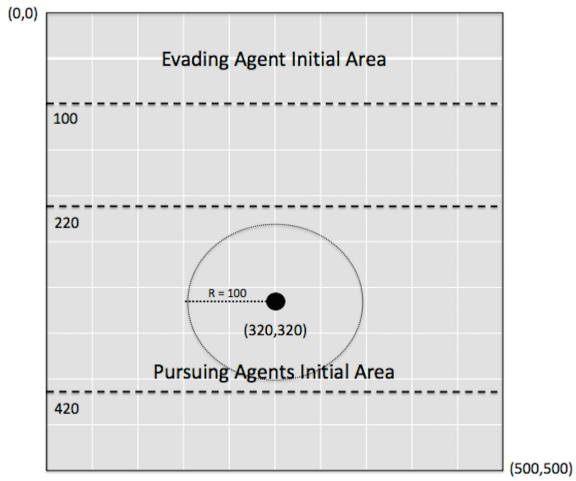

Figure 1: Grid layout of simulation environment. Both initial position boundary rectangles for evading and pursuing agents are shown. The middle ellipse portrays the orbital radius followed by agents invoking the *circle* role.

center. This created a simple evasion technique. Pursuing agents possessed three possible roles: $R_p = \{patrol, circle, pursuit\}$, with the following actions, $A_p = \{separate, seek, orbit, signal\}$.

Roles:

1. *patrol* – random movement around the grid.
2. *circle* – move in a circular pattern around the nest center.
3. *pursuit* – move toward evading agent.

Actions:

1. *separate* – steer away from other agents, otherwise, remain on course.
2. *seek* – move toward a coordinate.
3. *orbit* – move toward nest if current position is greater than 100 pixels away from nest center, otherwise, stay on course.
4. *signal* – send coordinates of evading agent to any pursuing agent within 200 pixels.

Algorithm 2 : Nest Defense

```
procedure ROLEASSIGNMENT
    changeRole(environmentState)
    executeActions(r_curent)
end procedure

procedure CHANGEROLE(STATE)
    for r ∈ allRoles do
        thresholdMet(r, STATE)
    end for
end procedure

procedure THRESHOLDMET(ROLE, STATE)
    if detectEnemy(state) or signal received then
        r_current ← pursuit return
    else if (state.distance.nest > 100) then
        r_current ← circle return
    else
        r_current ← patrol return
    end if
end procedure

procedure DETECTENEMY(state)
allObjectsInRange ← object.state.distance ≤ 100
    for allObjectsInRange do
        if object == enemy then return true
        end if
    end for
    return false
end procedure

procedure EXECUTEACTIONS(ROLE)
    if role == pursuit then
        signalNeighbor()
        seek evading agent
    else if (role == circle) then
        orbit + separate
    else
        separate
    end if
end procedure

procedure SIGNALNEIGHBOR
    for neighbors ∈ distance do
        signal
    end for
end procedure
```

Control and Independent Variables

Table 4 presents the experiments' control and independent variables. Simulations changed evading agent speeds, pursuing agent team sizes, and signaling actions. Figure 2 shows experiment construction. Using roles and signals as major independent factors, each block represents 20 experiments. For example, the first experiment set pursuing agent signal actions to true, roles to *patrol* and *pursuit*, and pursuing agent swarm size to 5, while setting evading agent speed to 0.50 pixels per time step. The second experiment incremented the pursuing agent swarm size to 10, the next experiment to 15, and so on. The next set of experiments increased the evading agent speed to 0.75 pixels per time step and pitted them against pursuing swarms of 5, 10, 15, and 20 agents. Overall, 80 experiments, each simulated 10,000 times, were performed.

5 Results

Tables 5 and 5 summarize pursuing agent win ratios as both the number of pursuing agents and evading agent speed increased. Table 5 holds the baseline results as agent roles were limited to patrol and pursuit (patrol-pursuit) with signaling enabled. Table 5 shows the results of expanding the number of roles to include the *circle* role (circle-pursuit). It is clear that both the number of pursuing agents and evading agent speed impact the performance of the pursuing swarm. Both types of pursuing agent swarms, patrol-pursuit

Control	
Nest Location	Coordinate (320,320)
Agent Visual Field	360 degrees
Pursuing Agent Speed	0.50
Agent Sensor Range	100
Agent Signal Range	200
Orbit Radius	100
Independent	
Evading Agent Speed	0.5, 0.75, 1.00, 1.25, 1.50
Pursuing Team Size	5, 10, 15, 20
Signal Action	On / Off

Table 1: Control and Independent Variables. All speeds are in pixels per time step, with ranges and radius length in pixels.

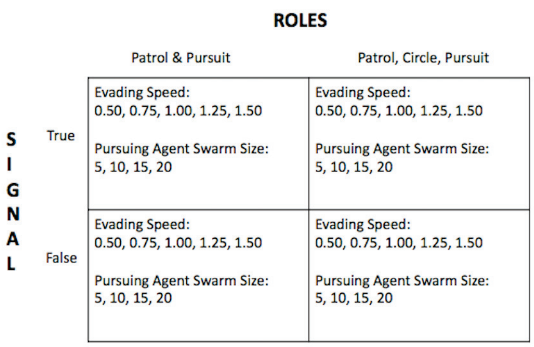

Figure 2: Experiments

and circle-pursuit, experienced a decrease in win percentages as the speed of the evading agent increased. Additionally, both types of pursuing agent swarms increased their win percentage as the number of agents in the swarm increased. Intuitively, these results make sense as the likelihood of detecting an evading agent will increase with the number of agents, and by extension the area of sensor coverage.

Evader Speed	Number of Pursuing Agents: 5	10	15	20
0.5	0.292	0.495	0.652	0.761
0.75	0.296	0.524	0.670	0.779
1.00	0.249	0.455	0.607	0.742
1.25	0.223	0.402	0.553	0.670
1.50	0.198	0.362	0.501	0.616

Table 2: Patrol-Pursuit agent win ratios versus evading agent speed.

To test the impact of signaling on swarm performance, signaling actions were removed from both patrol-pursuit and circle-pursuit swarms for comparison (Figures 3 - 6). For patrol-pursuit teams, the difference in performance is negligible across the range of tested values with the win ratio plots for both patrol-pursuit teams falling within the 95% confidence bound of each other. One could attribute this outcome to the random placement of the patrol-pursuit agents. The random placement made it difficult for agents to form sub-teams as agents were likely outside of signal range.

Evader Speed	Number of Pursuing Agents: 5	10	15	20
0.5	0.835	0.960	0.986	0.994
0.75	0.545	0.789	0.902	0.952
1.00	0.327	0.578	0.745	0.850
1.25	0.248	0.454	0.626	0.732
1.50	0.213	0.395	0.546	0.658

Table 3: Circle-Pursuit agent win ratios versus evading agent speed.

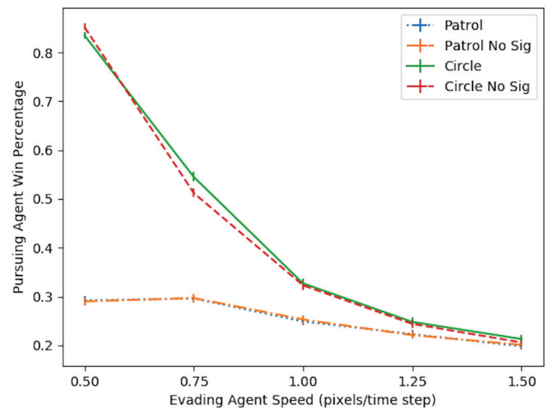

Figure 3: Win percentages for 5 pursuing agents pitted against a single evading agent.

The circle-pursuit teams faired better against evading agents across the majority of evading agent speeds and swarm sizes. Starting at a swarm size of 10, circle pursuit teams with signaling performed better than their non-signaling counterparts. The close proximity of agents to one another led to a higher probability of sub-team formation, resulting in swarm formations conducive to evading agent capture. Non-signaling teams experienced delays in reaction times as each agent had to detect the evading agent on their own. The signaling behavior created formations of sub-teams led by a temporary leader, a behavior noted in real-world pigeon flocks (Chen, Liu, and Zhang 2016). These formations created a net of pursuing agents which an evading agent found difficulty in escaping. These complex behaviors were obtained through simple role and signal actions.

Figure 7 presents a screen shot from a 1 versus 10 simulation. Four pursuing agents (boxed) have detected the evading agent, switched to the *pursuit* role, and are converging towards the evading agent's position. The other pursuing agents (not boxed) are engaged in either *patrol* or *circle* roles. The evading agent (circled) will be unable to avoid all four pursuing agents in this simulation. The novelty of the

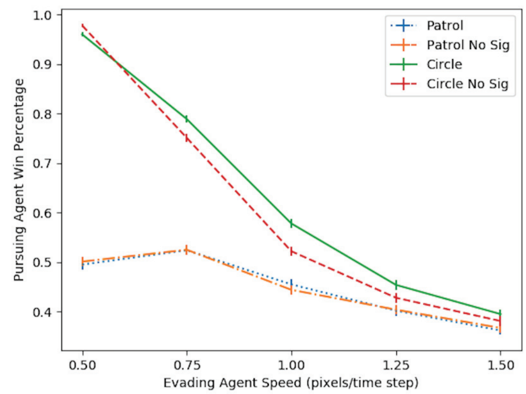

Figure 4: Win percentages for 10 pursuing agents pitted against a single evading agent.

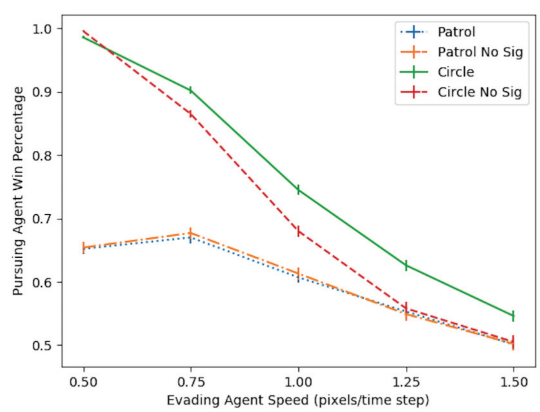

Figure 5: Win percentages for 15 pursuing agents pitted against a single evading agent.

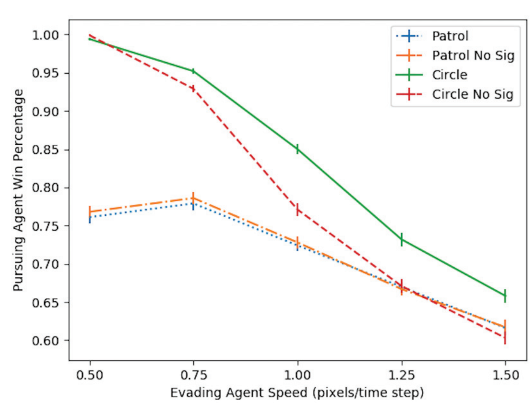

Figure 6: Win percentages for 20 pursuing agents pitted against a single evading agent.

ITAA is the emergence of these small collaborative teams without the need for a centralized control authority.

Finally, defending teams were tasked with preventing a multi-evader team from reaching the nest center. Multi-evader teams consisted of three evading agents using flocking rules to stay in formation. When encountering a defending agent, the multi-evader team would split apart in an avoidance maneuver while always moving towards the nest. Figure 8 provides the win rates for defending teams of 20 agents as they were the most successful in the single evader trials.

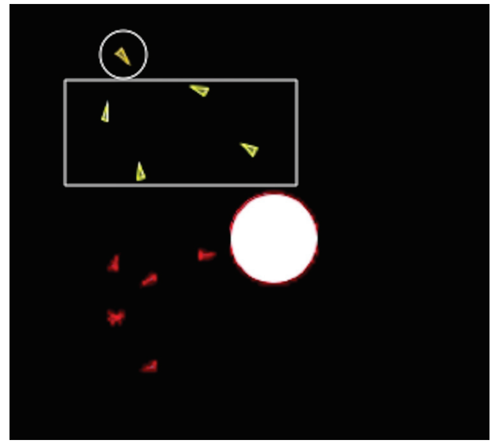

Figure 7: During this simulation, four pursuing agents in *pursuit* roles (boxed) converge on an evading agent (circled). Other pursuing agents remain in their current roles (*circle* or *patrol*). The white dot represents the nest.

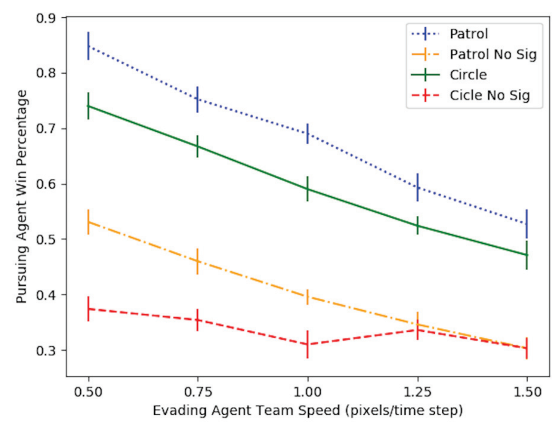

Figure 8: Win percentages for 20 pursuing agents pitted against three evading agents.

Surprisingly, in this scenario, patrol-pursuit teams with signaling faired better than any other team type. This outcome is likely due to team coupling, as in, patrol-pursuit agents dispersed into smaller teams across the defender area while circle-pursuit teams combined into larger, concentrated groups. These larger groups were effective at stopping one evader but in a multi-evader scenario, if an evader slipped by the large group it could continue, unfettered, to the nest. For patrol-pursuit teams, missing an evader was less disastrous as an evader would likely encounter another patrol-pursuit team. The signaling behavior for both types of teams clearly helped achieve higher win rates than non-signaling teams.

6 Conclusion

This paper presented the informal team assignment algorithm (ITAA). Experimental results showed that informal team assignment, based on local environmental conditions, positively effected the performance of a large swarm in a defensive position. In both single and multi-evader scenarios, the ability to change roles and alert neighbors to an evading agent's position resulted in higher capture rates over similar agents without signaling capabilities.

Results also showed that the ITAA enabled spontaneous creation of teams in response to environmental stimulus. These collaborative teams, making up sub-swarms, worked towards completing a shared goal. This behavior allowed an autonomous swarm to achieve multiple goals concurrently. In the multi-evader scenario, this behavior directly contributed to the higher capture rates achieved by signal-enabled pursuit teams.

Future work needs to address some shortfalls with the current approach. First, while the simulation approach is theoretically solid, mathematical models mapping actions to system level behaviors are lacking. A mathematical model could enable outcome predictions as different variables change (e.g. swarm increases, evading agent speed increases etc.). Furthermore, a detailed analysis of how many defenders are necessary to defend a certain region needs to be completed, allowing for more refined simulations and results. Finally, hybrid teams should be tested to see if mixed teams of circle and pursuit agents fare better in single and multiple evader scenarios.

References

Alexopoulos, A.; Schmidt, T.; and Badreddin, E. 2015. Cooperative pursue in pursuit-evasion games with unmanned aerial vehicles. In *Proceedings of 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.

Beard, R.; McLain, T.; Goodrich, M.; and Anderson, E. 2002. Coordinated target assignment and intercept for unmanned air vehicles. *IEEE Transactions on Robotics and Automation* 18(6).

Bedau, M. A., and Humphreys, P., eds. 2008. *Emergence: Contemporary Readings in Philosophy and Science*. MIT Press.

Bonabeau, E.; Dorigo, M.; and Theraulaz, G. 1999. *Swarm Intelligence: From Natural to Artificial Systems*. Oxford University Press.

Cao, L.; shun Tan, H.; Peng, H.; and cong Pan, M. 2014. Multiple uavs hierarchical dynamic task allocation based on pso-fsa and decentralized auction. In *2014 IEEE International Conference on Robotics and Biomimetics (ROBIO 2014)*, 2368–2373.

Chen, D.; Liu, X.-L.; and Zhang, H.-T. 2016. Switching hierarchical leadership in coordinated movement of pigeon flocks. In *Proceedings of the 35th Chinese Control Conference*.

Duan, H.; Li, P.; and Yu, Y. 2015. A predator-prey particle swarm optimization approach to multiple ucav air combat modeled by dynamic game theory. In *IEEE/CAA Journal of Automatica Sinica*, volume 2.

Holland, J. H. 1998. *Emergence: From Chaos to Order*. Addison-Wesley.

Holland, J. H. 2012. *Signals and Boundaries: Building Blocks for Complex Adaptive Systems*. MIT Press.

Kania, E. 2016. Swarms at war: Chinese advances in swarm intelligence. *China Brief* 17(9).

Li, D. 2006. *Multi-Player Pursuit-Evasion Differential Games*. Ph.D. Dissertation, The Ohio State University.

McCune, R.; Purta, R.; Dobski, M.; Jaworski, A.; Madey, G.; Madey, A.; Wei, Y.; and Blake, M. B. 2013. 2013 winter simulations conference (wsc). In *Investigations of DDDAS for command and control of UAV swarms with agent-based modeling*, 1467–1478.

Mitchell, M. 2006. Complex systems: Network thinking. *Artificial Intelligence* 170(18):1194–1212.

Reynolds, C. W. 1987. Flocks, herds, and schools: A distributed model. *Computer Graphics* 21(4):25–34.

Serugendo, G.; Foukia, N.; Hassas, S.; Karageorgos, A.; Mostefaoui, S.; Rana, O.; Ulieru, M.; Valckenaers, P.; and Aart, C. V. 2004. Self-organisation: Paradigms and applications. In *Engineering Self-Organising Systems: Nature-Inspired Approaches to Software Engineering*, 1–19. Springer.

Tan, Y., and Zheng, Z. 2013. Research advance in swarm robotics. *Defence Technology* 9(1):18–39.

Tolba, S.; Ammar, R.; and Rajasekaran, S. 2016. Taking swarms to the field: Constrained spiral flocking for underwater search. In *2016 IEEE Symposium on Computers and Communications (ISCC)*.

Vial, R.; Rashid, S.; Sharp, C.; Shakernia, O.; Kim, J.; and Sastry, S. 2001. Pursuit-evasion games with unmanned ground and aerial vehicles. In *Proceedings of the 2001 IEEE International Conference on Robotics and Automation*.