



**Emergent Behavior Development and Control in
Multi-Agent Systems**

DISSERTATION

David W. King, Maj, USAF
AFIT-ENG-DS-19-S-007

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

DISTRIBUTION STATEMENT A
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

The views expressed in this document are those of the author and do not reflect the official policy or position of the United States Air Force, the United States Department of Defense or the United States Government. This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

AFIT-ENG-DS-19-S-007

EMERGENT BEHAVIOR DEVELOPMENT AND CONTROL
IN MULTI-AGENT SYSTEMS

DISSERTATION

Presented to the Faculty
Graduate School of Engineering and Management
Air Force Institute of Technology
Air University
Air Education and Training Command
in Partial Fulfillment of the Requirements for the
Degree of Doctor of Philosophy in Computer Science

David W. King, B.S.C.S., M.S.C.O.

Maj, USAF

August 14, 2019

DISTRIBUTION STATEMENT A
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

AFIT-ENG-DS-19-S-007

EMERGENT BEHAVIOR DEVELOPMENT AND CONTROL
IN MULTI-AGENT SYSTEMS

DISSERTATION

David W. King, B.S.C.S., M.S.C.O.
Maj, USAF

Committee Membership:

Dr. Gilbert L. Peterson
Chair

Dr. Douglas D. Hodson
Member

Dr. Christine M. Schubert Kabban
Member

Abstract

Emergence in natural systems is the development of complex behaviors that result from the aggregation of simple agent-to-agent and agent-to-environment interactions. Emergence research intersects with many disciplines such as physics, biology, and ecology and provides a theoretical framework for investigating how order appears to spontaneously arise in complex adaptive systems. In biological systems, emergent behaviors allow simple agents to collectively accomplish multiple tasks in highly dynamic environments; ensuring system survival. These systems all display similar properties: self-organized hierarchies, robustness, adaptability, and decentralized task execution. However, current algorithmic approaches merely present theoretical models without showing how these models actually create hierarchical, emergent systems. To fill this research gap, this dissertation presents an algorithm based on entropy and speciation - defined as morphological or physiological differences in a population - that results in hierarchical emergent phenomena in multi-agent systems. Results show that speciation creates system hierarchies composed of goal-aligned entities, i.e. *niches*. As niche actions aggregate into more complex behaviors, more levels emerge within the system hierarchy, eventually resulting in a system that can meet multiple tasks and is robust to environmental changes. Speciation provides a powerful tool for creating goal-aligned, decentralized systems that are inherently robust and adaptable, meeting the scalability demands of current, multi-agent system design. Results in base defense, k - n assignment, division of labor and resource competition experiments, show that speciated populations create hierarchical self-organized systems, meet multiple tasks and are more robust to environmental change than non-speciated populations.

Table of Contents

	Page
Abstract	iv
List of Figures	viii
Acknowledgements	xiii
I. Introduction	1
1.1 Hypothesis	3
1.2 Background	3
1.3 Research Questions	6
1.4 Contributions	7
1.5 Limitations	8
1.6 Outline	8
II. Background	13
2.1 Emergence	13
2.2 Approaches	16
2.3 Definitions	18
2.4 Entropy	22
2.5 Genetic Algorithms	26
III. Informal Team Assignment Algorithm	29
3.1 Introduction	29
3.2 Background	30
3.3 Algorithm	32
3.4 Experiments	33
3.5 Scenario	33
3.6 Agent Implementation	34
3.7 Control and Independent Variables	36
3.8 Results	37
3.9 Conclusion	42
IV. A Macro-Level Order Metric for Complex Systems	44
4.1 Introduction	44
4.2 Motivation	45
4.3 Related Work	47
4.4 Approach / Methodology	51
4.5 Experimental Analysis	56
Autonomous UAV Simulation	56

	Page
Scenario 1	56
Scenario 2	58
Scenario 3	60
Game of Life	61
4.6 Discussion	65
4.7 Conclusion.....	67
V. Entropy-Based Team Self-Organization with Signal Suppression	69
5.1 Introduction	69
5.2 Motivation	69
5.3 Related Work.....	71
5.4 Problem Statement	74
5.5 Entropy-based Team Self-organization	75
5.6 Experiment	76
5.7 Results	77
5.8 Discussion	85
5.9 Conclusion.....	86
VI. The Emergence of Division of Labor in Multi-Agent Systems	88
6.1 Introduction	88
6.2 Motivation	89
6.3 Background	91
6.4 Methodology	94
Evolutionary Method	94
Domain	96
6.5 Experimental Analysis	97
Pin Factory	97
Environment	98
Simulations	98
Data Analysis	99
Division of Labor Experiments	99
Resiliency Experiments	106
6.6 Discussion	110
6.7 Conclusion.....	112
VII. A DEVS-Based Approach Towards Hierarchical Ecosystem Modeling	114
7.1 Introduction	114
7.2 Motivation	114
7.3 Background	116
7.4 Eco-DEVS Architecture.....	120
7.5 Example	124

	Page
7.6 Experiment	130
7.7 Conclusion.....	135
VIII. The Rise of Order: Emergence and Speciation in Multi-Agent Systems	137
8.1 Introduction	137
8.2 Motivation	138
8.3 Background	140
8.4 Approaches	143
8.5 Engineering Systems with Speciation	145
8.6 Speciation Algorithm (SEA)	147
8.7 Domains of Interest	150
Pursuit-Evasion Games	150
K Coverage Problem	153
8.8 Division of Labor	157
Pin Factory	158
Environment	158
Simulations	159
Experiment	161
8.9 Conclusion.....	164
IX. Conclusion and Future Work	165
9.1 Summary of Findings	165
9.2 Recommendations	167
9.3 Contributions to Knowledge	168
Bibliography	170

List of Figures

Figure	Page
1	Complex Adaptive Systems [57]. 15
2	Tiered structure for emergent systems. 19
3	Recursive Agent Formation. 20
4	Example genetic algorithm. An initial set of chromosomes (top of figure) run through a simulated environment. Afterwards, a fitness function assesses each chromosome's performance, selecting the best scoring individual's for breeding. Breeding is accomplished via the cross-over operation. Finally, each part of the chromosome is given a small probability of mutating into another form (e.g. switch from 0 to a 1). The process starts again with the new population and continues until encountering a pre-determined stopping condition. 27
5	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 <i>circle</i> role. 34
6	Experiments 37
7	Win percentages for 5 pursuing agents pitted against a single evading agent. 39
8	Win percentages for 10 pursuing agents pitted against a single evading agent. 39
9	Win percentages for 15 pursuing agents pitted against a single evading agent. 39
10	Win percentages for 20 pursuing agents pitted against a single evading agent. 40
11	During this simulation, four pursuing agents in <i>pursuit</i> roles (boxed) converge on an evading agent (circled). Other pursuing agents remain in their current roles (<i>circle</i> or <i>patrol</i>). The white dot represents the nest. 41

Figure		Page
12	Win percentages for 20 pursuing agents pitted against three evading agents.	41
13	Three targets (circles) detected by the swarm (triangles).	53
14	Swarm self-organizes into three teams which begin flying towards selected targets.	53
15	Macro-level order over time for UAV Scenario 1.	58
16	Macro-level order over time for UAV Scenario 2.	59
17	Macro-level order over time for UAV Scenario 3.	61
18	Macro-level order over time for R Pentomino.	63
19	Macro-level order over time for a Glider.	63
20	Macro-level order over time for a Gosper Gun.	64
21	Average detection rates across all sensor ranges with teams of n trackers and m observers (x-axis), $k = 3$	78
22	Average saturation rates across all sensor ranges with teams of n trackers and m observers (x-axis), $k = 3$	79
23	Correlation between detection and k-n coverage for mixed teams of 5 trackers and n observer agents across tested sensor ranges, with $k = 3$	82
24	Correlation between detection and k-n coverage for mixed teams of 10 trackers and n observer agents across tested sensor ranges, with $k = 3$	83
25	Correlation between detection and k-n coverage for mixed teams of 15 trackers and n observer agents across tested sensor ranges, with $k = 3$	83
26	Correlation between detection and k-n coverage for mixed teams of 20 trackers and n observer agents across tested sensor ranges, with $k = 3$	84
27	Correlation between detection and k-n coverage for mixed teams of 25 trackers and n observer agents across tested sensor ranges, with $k = 3$	84

Figure		Page
28	Average number of pins manufactured for 1:1 scenarios using the fixed threshold model (FTM) and linear threshold model (LTM).	103
29	Average number of pins manufactured for 3:1 scenarios using the fixed threshold model (FTM) and linear threshold model (LTM).	104
30	Average number of pins manufactured for 5:1 scenarios using the fixed threshold model (FTM) and linear threshold model (LTM).	104
31	Average number of pins manufactured using the fixed threshold model (FTM) at 1:1, 3:1, and 5:1 agent to task ratios with a change in environment at generation 1,000.	108
32	Average number of pins manufactured using the entropy-based linear threshold model (LTM) at 1:1, 3:1, and 5:1 agent to task ratios with a change in environment at generation 1,000.	108
33	Notional system diagram using eDEVS structure. The dotted lines represent hierarchical niches with strict input and output signals specified by directional arrows. In this particular diagram, messages and global variables represent the message signaling techniques employed by niche subcomponents.	124
34	Object-oriented view of agent models.	130
35	Colony A strategies over time. Colony A begins with the superior ‘Rock’ strategy and experiences an increase in agent population. However, the population begins to decline with Colony B switches to the ‘Paper’ strategy. Eventually, Colony A chooses the ‘Scissors’ strategy, eventually eliminating Colony B’s population. Here one sees the dynamics between macro-level guidance and micro-level decisions as Colony A’s strategic changes influenced agent-level interactions, resulting in a positive outcome.	131

36	Colony B strategies over time. Although Colony B starts the simulation with a disadvantageous strategy, it switches to a superior at time step 20. Colony B then experiences a population explosion, almost leading to the destruction of Colony A. However, it experiences a population decline when Colony A switches strategies. Interestingly, although Colony B holds the inferior strategy, it can maintain its population. In response to Colony A's population increase, it changes to the superior strategy of 'Rock' and begins to impact Colony A's population but only for a short time. As the strategy does not increase its overall population, Colony B begins oscillating strategies at time step 140, only to succumb when it chooses the inferior strategy 'Paper' when Colony A holds a distinct population advantage and strategy.	132
37	Colony populations overtime for competing strategies. Clear predator-prey oscillation patterns emerge as both colonies vie for supremacy in their environment. The 90-degree offsets for the populations mirror the expected predator-prey oscillations predicted by Lotka-Volterra equations, showing how the eDEVS framework can capture emergent behaviors in complex adaptive systems.	133
38	Colony A and B population transitions from time step 60 to 70. Lighter squares represent higher population densities. Although not implemented in the system, a hunting behavior emerged as a clear movement towards Colony B's population occurs when Colony A changes to the 'Scissors' strategy at time step 40 and agents receive an increase in their aggressive stimuli from the macro-level.	134
39	Complex Adaptive Systems [57].	142
40	Updated view of a CAS.	146
41	Updated world view. Agents at the micro-level align (1) to functional communities (2) which can combine to form macro-level emergent behaviors (3). Feedback flows up and down the hierarchy, creating changes in functional niches, which then creates macro-level change, in an eternal cycle of causal feedback loops.	147

Figure		Page
42	During this simulation, four pursuing agents in <i>pursuit</i> roles (boxed) converge on an evading agent (circled). Other pursuing agents remain in their current roles (<i>circle</i> or <i>patrol</i>). The white dot represents the nest. [73]	153
43	Average detection rates across all sensor ranges with teams of n trackers and m observers (x-axis), $k = 3$. [74]	156
44	Average saturation rates across all sensor ranges with teams of n trackers and m observers (x-axis), $k = 3$. [74]	157
45	Average pins produced by three, evolutionary populations. Entropy populations used the LTM, while non-entropy populations used the FTM.	162
46	Average pins produced by three, non-evolutionary populations. Entropy populations used the LTM, while non-entropy populations used the FTM.	163

Acknowledgements

If a mosquito has a soul, it is mostly evil. So I don't have too many qualms about putting a mosquito out of its misery. I'm a little more respectful of ants. – Douglas Hofstadter

When many work together for a goal, great things can be accomplished. It is said a lion cub was killed by a single colony of ants. – Saskya Pandita

This work, like many things in life, is an accomplishment of many individuals working together. Many thanks to my committee and my family for continued support throughout this effort.

David W. King

EMERGENT BEHAVIOR DEVELOPMENT AND CONTROL IN MULTI-AGENT SYSTEMS

I. Introduction

Curiosity drives humankind towards navigating the deepest parts of the earth's oceans to capturing glimpses of the universe's earliest light. In between these vast expanses exist ecosystems of various structures and complexities. Residing within each are numerous adaptive organisms and particles, each creating their own microcosms full of wondrous, and mysterious, behaviors. These complex adaptive systems (CAS) surround us, impacting everything from how well one's lawn grows to global weather patterns. These natural systems generate high research interest because of their inherent properties: hierarchical structure, robustness, adaptability and decentralized execution of tasks; constructed upon a base of relatively simple components [13, 55, 57, 93, 96, 103, 108, 103].

In today's world, the fielding of multi-agent systems composed of autonomous entities is rapidly increasing [25, 49, 68]. These systems must be robust to agent loss, adaptable to environmental change, and possess the ability to concurrently complete multiple tasks. The high complexity of agent interactions, possible system states, and volume of agent communication and sensor data overwhelms centralized control approaches [17, 91, 129]. Additionally, like their natural counterparts, these systems contain non-linear behaviors that are difficult to predict and control [25, 49]. These behaviors, referred to as emergent [14, 55], create a challenging design problem.

The problem facing engineers is how to design a complex, multi-agent system that is inherently decentralized yet adaptable and robust. One way to do this is by developing a self-organizing system that has an emergent property matching a system goal, i.e. a desired

macro-level behavior. However, an agreed upon framework, or model, for engineering such systems currently eludes the field [69, 57, 30]. Emergence researchers contend that emergent systems possess beneficial characteristics such as self-organizing behaviors, robustness, adaptability, and decentralized control [13, 55, 57, 93, 96, 103, 108]; all desired characteristics for engineered systems. While many modeling approaches exist [57, 103, 49], relatively few [33] look at the impact of agent physiology or morphology upon emergent behaviors. This dissertation formulates multi-agent, emergent systems based upon one major concept: speciation, and studies its role in the evolution of weak emergent phenomena to achieve a macro-level goal.

Speciation is the diversity of agents in a population. Differences can be *morphological* or *physiological*. Morphological differences are physical differences such as larger body size and mandibles that differentiate soldier and worker castes in some ant colonies [139]. These physical differences imply forced speciation through long-term evolutionary processes. For engineered systems, human-derived differences such as various sensor suites and engines replace natural evolution processes with human-led ones. The development of different types of vehicles and refinement of their capabilities is still evolutionarily iterative, just artificial.

Physiological differences refer to how the agent functions, and changes based on how the agent digests information from its environment [4]. For example, the social spider *Anelosimus studiosus* all look the same, and all have the same sets of skills, e.g. web maintenance, hunting, brood care, etc. However, Holbrook, et al. [53] found that docile and passive members of this particular species engage in particular tasks aligned with temperament, with aggressive spiders trending towards hunting while passive ones took on web maintenance and brood care tasks. These differences lie within the spiders, and could be based on genetic material or learned experiences. The end result is physiological speciation. One primary trait of physiological speciation of interest here is the ability of an agent

to change its function based on the current environment or learned experiences. This makes physiological speciation especially powerful as it creates agents with plasticity.

Biologically speaking, plasticity is where an agent adapts, either morphologically or physiologically, to changes in its environment [5]. Delving further into the habits of the social spider *Anelosimus studiosus*, Holbrook, et al. [53] found that if they founded a colony of all aggressive, or all passive, spiders, the colonies self-organized into groups that carried out all the required tasks, e.g. web maintenance, hunting, etc., regardless of the spider’s internal bias. In other words, the spiders showed physiological plasticity by accomplishing less desirable tasks, ensuring colony survival.

1.1 Hypothesis

This work hypothesizes that the combination of speciation with the beneficial characteristics of emergent systems, results in a generalizable, decentralized-control methodology for complex, multi-agent systems. At the agent level, speciation aligns agents into functional groups, defined as *niches*. Each niche aligns towards a sub-goal, creating a hierarchical level composed of different niche *species*. As niche groups work towards completing sub-goals, their actions aggregate into an emergent macro-level behavior that matches a system goal. By using speciation, one can engineer a multi-level, hierarchical system whose inherent emergent behaviors support the system’s overall purpose.

1.2 Background

Emergence research intersects with many disciplines such as physics, biology, and ecology [56, 93, 97]. It provides a theoretical framework for those investigating how order appears to spontaneously arise in complex adaptive systems [70]. Order in these systems is associated with complex, collective behaviors that emerge from the nonlinear interactions of entities with one another and their environment [128]. For Reynolds’ boids [110],

organized flocks emerged from the chaos of a mass of birds moving through the sky. For Conway [116], extraordinary patterns of *life* grew from simple rules on a plane. In both examples, order arose from agent interactions in both continuous and discrete domains, created and restricted by simple interaction rules. The interplay between local agent interactions, i.e. micro-level actions, and global outcomes, i.e. macro-level behaviors, serves as the fundamental concept behind emergence theory.

One of the primary issues with emergence is the philosophical debate that surrounds it based upon the deceptively simple question, “what is emergence?” In other words, when one observes a *surprising* behavior in a natural or engineered system, is it an emergent behavior or simply resultant [9]? Depending on whom one asks, the answer is yes or no and the debate dates back to the 1800s when G.H. Lewes noted that certain natural phenomena could not be expressed in quantitative terms [24]. These phenomena appeared to exceed the simple summation of the parts and components which comprised them [55]. Lewes felt these were *emergent* behaviors and advocated they may be unexplainable.

The idea of unexplainable phenomena does not sit well with most in the scientific community. Accepting that these behaviors are unexplainable makes the study of emergence fairly pointless, and as Hempel and Oppenheim [52] state, “it encourages an attitude of resignation which is stifling to scientific research.” This argument forces emergence researchers to find a compromise, or give up the pursuit altogether.

Although the community lacks a full consensus on emergent behavior, generally speaking, an emergent behavior is a novel, or surprising behavior pattern displayed by a complex system [112]; that cannot be explained as merely the sum of its parts [103]. However, the words *surprising* and *novel* complicate the definition. For example, flocking is a commonly accepted emergent behavior [112, 24, 9]; however, Reynolds [110] invented three simple rules for moving objects, called *boids*, that produce realistic simulations of birds flocking. If one programmed a few aerial drones with Reynold’s method, then one would expect flock-

ing to occur, it would actually be *surprising* if it did not. Does this mean flocking is no longer emergent? In the author’s opinion, no. Flocking is still an emergent property as it is still a macro-level behavior that emerges from the interactions of simple agents. This leads to Chalmers’ definition: “Emergence is the phenomenon wherein complex, interesting high-level function is produced as a result of combining simple low-level mechanisms in simple ways” [19]. This definition avoids using terms such as *novelty* and *surprise* while allowing for causation and prediction arguments.

Both causation and prediction are important for those seeking to build emergent behaviors into engineered systems. Without causation one could not create a system with desired emergent properties, and, although prediction is still very difficult in dynamic environments as it is impossible to know all the unknown variables associated with the system and its environment, having some sense of how the system will behave, e.g. robots will flock, build a path based on pheromones, etc., is a nice property. This leads to Bedau’s solution to the emergence problem. Bedau [9] defines two primary types of emergent behaviors: weak and strong. Weak emergence is a behavior or pattern that can be reduced to actions and behaviors of agents at the micro-level. Strong emergence, on the other hand, is irreducible, meaning the causal pathways, up or down, cannot be ascertained. As this work is concerned with real-world systems, it accepts both Chalmer’s and Bedau’s definitions as they facilitate the application of emergence research.

This dissertation focuses on providing models and algorithms that can be applied in engineered systems that allow one to design systems with desired emergent behaviors, e.g. division of labor, system robustness, etc., without the need for a centralized controller, or necessitating heavy heuristic guidance formulated by subject matter experts. In other words, design systems that are highly adaptable and robust *online*, with minimal outside guidance, based upon properties found in emergent systems with a specific interest on the impact of agent speciation, both morphological and physiological.

1.3 Research Questions

The primary research hypothesis is that speciation creates functional niches that establish a hierarchical self-organized system. These systems, designed from a goal-driven perspective, display beneficial emergent behaviors such as robustness and adaptability. The research questions presented here address different parts of the hypothesis.

1. Does agent speciation lead to hierarchical self-organized systems?
2. Do these systems show robustness in dynamic environments?
3. Can these systems accomplish multiple tasks concurrently free from a centralized control authority?

These questions all tie directly to theories proposed in emergence research. The vital difference here is this dissertation tests how agent speciation can create systems that possess these properties.

Research question one addresses a fundamental idea in emergence research: micro-level interactions formulate macro-level behaviors, thus imposing a hierarchy of relationships upon the system. Chapters III - VI each present a different method of instantiating speciation in agent populations engaged in a variety of domains. Results from each chosen domain show that speciated agents functionally align according to their morphology or physiology, resulting in the first level of self-organized behavior. These behaviors then aggregate into macro-level behaviors that align with system goals.

Research question two tests how robust these systems are under severe environmental change. Robustness means the system is able to recover back to its original state after a catastrophic environmental change or stabilize to a new state without collapsing altogether. Although each chapter shows systems robust to minor changes, Chapters VI and VIII explore the robustness of speciated populations under extreme environmental change, such as resource starvation, and how the system achieves a new steady state afterwards. Chapter IV presents a macro-level metric that graphically depicts the system state allowing for analysis

of system robustness and possible emergent phenomena. Chapter VIII explicitly evaluates the hypothesis itself.

Finally, research question three assesses if agent speciation leads to the emergence of behaviors such as division of labor that contribute to broad task capabilities. Although division of labor serves as the primary focus of Chapter VI, Chapters III, IV, and V all investigate systems which can only meet macro-level goals through self-organized, multi-task capabilities.

1.4 Contributions

The research presented adds speciation as a key component of emergent behaviors in engineered systems. Results show that speciation leads to a horizontal shifting of agent niches, i.e. groups of functionally similar agents. These micro-level shifts modify macro-level behaviors which, in turn, influence micro-level decisions, in a perpetual, feedback cycle. Results also show how speciation leads to system robustness and adaptability. As macro-level behaviors rely on micro-level interactions, this work highlights the power of decentralized execution of agent decision making with respect to both morphological and physiological constraints.

Overall, experiments show that speciation is key to building robust and adaptable, multi-agent systems. Agents guided by their physiology self-organize into functionally aligned niches which creates macro-level, emergent patterns. Emergent patterns such as the division of labor rely on, and influence, the appearance, growth, and disappearance of these niches through a feedback loop that cycles through the hierarchical levels of the system. The addition of horizontal movement, i.e. the population shifting between functional niches, adds an overlooked nuance to emergence research. Instead of the traditional, vertical feedback loop cycle found in models of emergent, hierarchical systems, this work shows how the interplay between agents, and even emergent behaviors, at each level cre-

ates, reinforces, and destroys macro-level emergent behaviors.

1.5 Limitations

Holland wrote that simulated models “describe possibilities not actualities [56]” and this work is representative of this fact. The various simulations in this work provide what could occur in engineered systems armed with the properties laid out through the following chapters. However, these are just possibilities. The number of unknown variables in real-world environments makes it virtually impossible to account for all behaviors, emergent or otherwise, that may occur. Additionally, these methods only apply to engineered systems. How speciation occurs in natural systems is as much a mystery now as it was when this research began. Although the author could argue that the concept of speciation could help with modeling natural behaviors, it does not provide reasoning for their occurrence. Furthermore, this work restricted itself to domains and systems with set descriptions and laws [55], enabling the production of artificial systems teeming with simple agents abiding by known rules and functions avoiding the philosophical minefield of *strong* emergence.

1.6 Outline

Each chapter explores the underlying premise of emergence research: simple agent interactions create complex, macro-level patterns. Thematically, each chapter incorporates a species of simple agents, with defined interactions, desired end state behavior patterns and decentralized control approaches. Although each chapter shares these traits, each is a stand alone work, with its own particular problem domain and methodology with the exceptions of Chapters II and IX.

Chapter II consists of background material, exploring the history of emergence research, establishing definitions for emergence, and related terms, as well as background related to methods used in later chapters. Readers familiar with emergence, Shannon’s in-

formational entropy and genetic algorithms, can readily skip this chapter as each follow-on chapter has its own related background and methodology sections.

Chapter III introduces the Informal Team Assignment Algorithm (ITAA). The ITAA shows how agent communication and speciation, defined as specific agent roles, lead to the dynamic formation of teams dedicated to a common goal: nest defense. This chapter aligns with the second research question about speciation and its impact on system robustness. Furthermore, as agents must accomplish multiple tasks concurrently, i.e. search and capture, this chapter supports research question three. The predefined roles with internal threshold mechanisms presented in this chapter represent physiological differences in the agent population and how they create desired emergent behaviors. In the patrol-pursuit domain presented in this chapter, pursuit teams spontaneously perform flanking maneuvers, which were not formally programmed into the system, to capture invading agents that possess superior speed and maneuver capabilities. Additionally, through the use of predefined roles, agents are capable of performing multiple tasks, e.g. patrol and pursuit, simultaneously without centralized control, producing both an adaptable system and a scalable control solution.

In Chapter IV, Shannon's informational entropy serves two purposes. First, agents in a 2D, multi-UAV domain use entropy to guide target selection decisions, ensuring an even distribution of agents across known targets. Here, agent speciation is achieved in an adaptive manner as agents continuously update their world view, and by extension their function, based upon current, local information. Second, Shannon's entropy is used to create a macro-level metric of system state. Used to measure the system state of the 2D, multi-UAV domain, and Conway's Game of Life [116], results show that the metric accurately depicts systems in steady states, as well as showing systems undergoing critical changes. These critical changes could be indicators of potential emergent phenomena, providing a possible method for identifying the formation of new system behaviors. The detected changes

also indicate the establishment of local niches, i.e. the appearance of hierarchical levels, in the system. This chapter explores research questions one and three, specifically showing how speciated populations, using entropy to guide agent decision, create hierarchical self-organized systems with multi-task capabilities.

Chapter V extends the use of entropy further in the k -coverage domain. In the k -coverage problem, a system must assign k agents across all known targets [104]. The difficulty in the domain lies in the fact that the number of targets and their locations are unknown *a priori*, and the agents have limited communication ranges. Although the entropy-based approach in Chapter IV provides a starting point, the desired end state is not a uniform distribution of agents across all n targets. This work elected to implement signal suppression techniques found in bee colonies to solve this problem. In bee colonies, bees can suppress the signals of other bees when it comes to recruiting others for resource gathering [107]. Agents mimic this behavior when they are closer to the target or when the target reaches a desired saturation rate, i.e. k agents are currently engaged with a target. The signal suppression technique ensures only k agents engage a target at each time step. Results show that the range of the suppression signal directly impacts the ability of the multi-UAV swarm to maintain the desired k -coverage rate. Again this chapter highlights how distributed decision making by simple agents can solve complex problems. Furthermore, this chapter builds morphological species of agents, called *observers* and *trackers*, to achieve k -assignments. This chapter compares the performance of true heterogeneous teams, i.e. those with morphological differences, and teams composed solely of one species who change their physiological response thresholds as the environment changes. By doing so, this chapter aligns with research questions two and three as the system under study must adapt to dynamic change as well as complete two tasks concurrently: detection and coverage.

Chapter VI investigates multi-agent system properties that lead to the emergence of

division of labor. It empirically studies the effects of agent decision functions, population size, and environmental constraints on the evolution of division of labor in a homogeneous population working in Adam Smith's Pin Factory [120]. Results show that both the choice of agent decision function and population size have strong correlations to the emergence of division of labor. However, it also shows that high degrees of division of labor, i.e. highly specialized agents, do not necessarily yield higher resource production, indicating that when designing emergent systems one must think critically about the ultimate system outcome. Additionally, as environmental constraints effect agent performance, the choice of the correct agent decision function also comes into play. This chapter highlights the power of adaptive physiology and how they evolve complex, adaptive populations, leading to the emergence of division of labor in the system under study. This chapter supports all three research questions as speciated populations must complete nine different tasks in a dynamic environment. The functional alignment of agents towards specific tasks creates niches that form desired, macro-level behaviors, in this case, the making of a resource. In experiments on robustness, populations undergo severe environmental change during runtime. Results show that speciated populations are both robust to environmental change and capable of completing multiple tasks concurrently.

Chapter VII provides a culminating chapter in modeling emergent systems. The modeling framework, called ecosystem discrete event simulation (eDEVs), incorporates portions of multi-level DEVs (ml-DEVs) established by Uhrmacher *et al.* [131], dynamic structure DEVs (DSDEVs) by Barros [6] and Holland's Echo model [54], to produce a hierarchical modeling framework with tight restrictions on information sharing. eDEVs considers each hierarchical level as a coupled model, defined as a *niche*, that conducts and constrains the types of signals and information that may pass through them. By logically separating levels into respective niches, the framework enables dynamic object switching, allowing for system adaptation. Using eDEVs, one can simulate complex behaviors based upon inter,

and intra, niche signaling. Additionally, eDEVS provides for temporal analysis and testing of mid and long term system goals, generalizing eDEVS' applicability. Tested in a simple ant colony domain, this chapter shows how eDEVS models adaptable emergent systems can allow for causal analysis of unanticipated emergent behaviors such as the formation of hunting parties. eDEVS is a framework that is capable of modeling hierarchical systems as well as micro-macro and macro-micro level interactions, generating behavior causation linkages missing in the majority of formal modeling approaches of emergent systems.

Chapter VIII formalizes the impact of agent speciation in emergent systems and evaluates the hypothesis itself. This chapter incorporates themes, definitions and approaches found in emergence research and shows how the incorporation of agent speciation creates robust, hierarchical self-organized systems with multi-task capabilities. Furthermore, a conceptual algorithm based on speciation is described and implemented in multiple domains. This work's hypothesis is supported by results from experiments in three different domains: pursuit-evasion games, the k -coverage problem, and Adam Smith's Pin Factory. Results show that speciated populations create hierarchical self-organized systems, meet multiple tasks and are more robust to environmental change than non-speciated populations. The use of the algorithm across multiple domains shows the generality of agent speciation and physiological adaptation, and adds speciation as a foundational part of emergent systems.

Finally, Chapter IX finishes the work covering avenues for future research opportunities and domains as well as concluding the work.

II. Background

Central to the primary hypothesis is showing how micro-level adaptations, driven by agent speciation, create goal-aligned *niches* in complex systems which, in turn, produce multi-task capabilities. In emergence theory, one finds many of the threads found throughout this work: micro and macro-level interactions, self-organization, hierarchical structures, and adaptation. This chapter covers the main research topics and tools used to answer the primary research questions.

Sections 2.1-2.3 cover the history of emergence, current approaches, themes and definitions. Section 2.4 reviews Shannon’s entropy and current entropy-based approaches and their use in guiding agent decisions in complex systems. These concepts feature prominently in Chapters IV, V and VI as entropy-based agent decision processes are shown to create *controllable* emergent patterns. Section 2.5 covers genetic algorithms. Chapter VI shows how division of labor can emerge by physiologically evolving entire agent populations using genetic algorithms.

2.1 Emergence

Emergence, and the debate surrounding it, dates back to the late 1800s when G.H. Lewes noted certain natural phenomena could not be expressed in quantitative terms [24]. Lewes was attempting to explain how complex behaviors, such as birds flocking, could not be directly correlated to the sum of its parts. In other words, if one understood a bird and all its potential actions, e.g. hopping, chirping, flying etc., one would not guess that dozens of flying birds could self-organize into a cohesive flock that flew at the same speed, in the same general direction, without running into one another and without a leader coordinating their actions. Lewes termed these types of complex behaviors *emergent*. Emergence fell out of favor as breakthroughs in genetics made emergent theories about evolutionary behaviors

inadequate as genes began explaining parts of human physiology [24]. However, in the late 1960s, emergence research reignited when Roger Sperry [24] coined the term *downward causation* to describe the functioning of the human brain, gaining momentum as Minsky [92] explored his theory of mind.

Broadly speaking, emergent behaviors are novel patterns displayed by complex systems. For example, a simple ant cannot find the shortest path between a food source and the nest. However, a million ants can construct such a path. This complex behavior, shortest path building, is displayed by the complex system, i.e. the ant colony, but not the individual ant. This line of thought results in two ideas: complex systems are composed of hierarchical levels with independent properties, and micro-level actions result in macro-level behaviors. The latter idea infers possible causation arguments, i.e. certain micro-level actions should create specific macro-level behaviors, while the former implies some self-organizing power inherent to natural, complex systems.

Bedau defines two forms of emergent behavior: weak and strong [9]. Weak emergence is a behavior that can be reduced to actions and behaviors of agents at a lower level. Furthermore, weak emergent systems display upward causation tendencies where micro-behaviors effect macro-behaviors [9]. Strong emergence, on the other hand, is irreducible. Some [9, 28] believe human consciousness is an example of *strong* emergence as it is not deducible by current knowledge [28]. As such philosophical debate is beyond the scope of this dissertation, work instead focuses on proposed *weak* emergent behaviors as they are hypothetically reducible, and therefore, one is able to reproduce them with simulated models.

One primary advocate of emergence research is John Holland. His works, *Hidden Order* [54], *Emergence: From Chaos to Order* [55], and *Signals and Boundaries* [57], provide theoretical models and ideas on how emergence occurs. Holland proposes that emergent patterns rely on the atomic actions of agents as well as the behaviors (or interac-

tions) between them [55] and models for such agents need to contain both (see Figure 1). Additionally, Holland states that the environment plays a role in the development of agent behaviors and interactions. The environmental argument is by Dautenhahn [27] in work about socially intelligent agents. Specifically stating that “intelligence can only be studied with a complete system embedded and coupled to its environment” and such agents must be adapted to their environment.

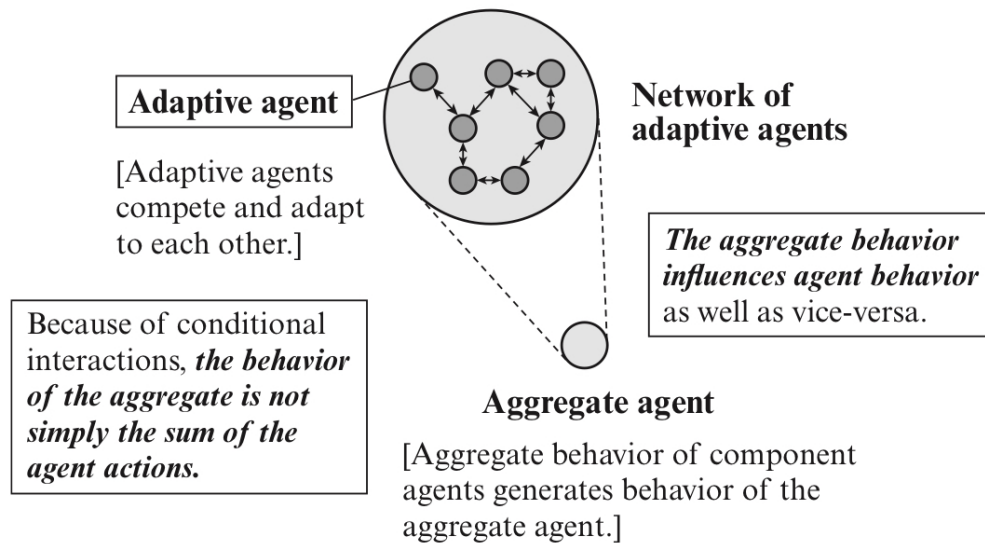


Figure 1. Complex Adaptive Systems [57].

Corning extends both arguments with work on *synergy*. According to Corning, emergent behavior is the result of agents within the system, as well as the environment, aligning in *purposeful* or *synergistic* ways, producing both expected and novel behaviors [24]. Novel behaviors arise as agents react to environmental stimuli *at every level* of the system. A complex system should display emergent behavior if, and only if, there exist synergies between agents at various levels. Additionally, the environment in which the agents *live* guides agents with continuous feedback. Feedback enables the agents, and by extension the system, to learn and adapt.

Emergence research theorizes the following about complex systems. One, emergent

systems consist of nested hierarchies built upon agent actions and interactions. Arguments for and against downward causation and exclusivity aside [9, 72, 61, 24], there is a prevailing belief that a hierarchical system relies upon macro-level behaviors emerging from micro-level interactions [13, 55, 57, 93, 96, 103, 108]. As one moves up the hierarchy, the behaviors encountered grow more complex, eventually forming a macro-level behavior that exceeds the capabilities of individual entities in the system [108]. Two, the building of complexity in the system makes it more flexible, and therefore robust to environmental change [7]. Flexibility hinges upon the notion of internal feedback loops. Macro-level behaviors influence and rely on changing micro-behaviors [55]. Finally, this view imposes a hierarchy of behaviors on the system where cyclic feedback loops allow the micro to influence the macro which, in turn, influences the micro, possibly creating a new macro-level pattern, or extending or ceasing an old one [57].

2.2 Approaches

There exists a high interest in predicting and controlling emergent behavior in many fields ranging from biology and ecology to systems engineering and computer science, with a particular focus on mapping micro to macro-agent level behaviors. The majority of research centers on desired end-state prediction instead of studying the impact of micro-level interactions. For example, de Wolf, et al. [142] avoided tying micro-interactions to macro-behaviors by building a goal driven system with little concern for causality. In other words, they designed the system from the top-down, versus the bottom-up. Morandini, et al. [96] used a similar approach, again building a goal driven system; however, they allowed agent adaptation to build a more robust system. Morandini, et al.'s method acknowledged the link between micro and macro-agent behaviors, but, like de Wolf, et al., they avoided investigating these causal links. These approaches assume a reductionist view of micro and macro-agent level behaviors focusing on downward causation, i.e. macro-level

goals guide micro-level interactions, ignoring benefits of tying micro-level interactions to macro-level goal accomplishment. Only Breddin [15] seems to advocate a systematic, top-down methodology where one studies a macro-level behavior and begins mapping out the components that lead to it's emergence. Their method at least acknowledges the role of micro-level interactions.

Pais [103] and Gignoux, et al. [41] used graph theory to map micro to macro-agent level interactions. Pais [103] used replicator-mutator dynamics (RMD) to analyze how some behaviors become dominant in social networks. Using a set of differential equations, RMDs model various strategies employed by agents inside the population and provide a method for offspring to inherit “fit” strategies as well as allow for mutations as seen in many genetic algorithms [62]. RMDs provide a reasonable method for mapping local-interactions to a global state (i.e. strategy). However, it is unclear how scalable the solution is for systems that possess more than three global strategies.¹ Their work also concentrated on known global strategies found in nature, knowing the constituent parts *a priori*.

Gignoux, et al. [41] presented a graph theory based model and ontology to study the interactions between micro and macro-agent level behaviors. They modeled agents as nodes in a network with relationships defining the edges between them. In this manner, one can construct a visual topography of the system. The authors admit one major drawback to their model is the maintenance of the edge relationships. Adding and deleting edges is cumbersome and has the potential for erroneous links remaining in the system or even accidentally removing beneficial links [41]. Their network based approach arises from both Holland [57] and Mitchell [93] who proposed that network theory could play a role in understanding how complex systems evolve.

A common theme for these approaches is the use of simulation as a substitute for formal mathematical proofs. Relying on the law of large numbers, running a large number of

¹Pais used a triangular simplex to graphically show the system shifting to one favored global strategy over two others [103].

simulations guarantees convergence to an average global state. As an example, Bonabeau, et al. [13] used Monte-Carlo simulations to produce close approximations of army ant raid patterns. Pais [103] and de Wolf [142] used simulation to verify their respective hypotheses with respect to global outcomes. By extension, one could expect that simulations could produce expected local outcomes (i.e. emergent behaviors below the macro-agent level) as well as global ones. The key issue is mapping agent and environmental interactions to those outcomes.

2.3 Definitions

Generally speaking, emergence is a novel, or surprising, behavior pattern displayed by a complex system [112]; that cannot be explained as merely the sum of its parts [103]. At the very core of emergence research is the idea of *surprise*. The behaviors displayed by the complex system *surprise* the observer and defy explanation. As there is some debate on the idea of *surprise*, this work adopts Chalmers' definition instead. "Emergence is the phenomenon wherein complex, interesting high-level function is produced as a result of combining simple low-level mechanisms in simple ways" [19]. Chalmers' definition avoids using terms such as *novelty* and *surprise* while allowing for causation and prediction arguments. He also incorporates another tenant of emergent study: agent interactions.

Many researchers believe that the interactions between agents, e.g. ants, molecules, etc., create complex emergent behaviors [55, 93, 117] inside a hierarchical structure. The term *agent* describes any entity inside a complex system that is a whole component (e.g. an ant or bee in a colony). These components and their interactions create hierarchical levels inside the environment.

In Figure 2, micro-level interactions combine to build complex, macro-level behaviors. These behaviors, in turn, combine to form other macro-level behaviors. Importantly, the structures and behaviors found at higher levels of the system do not exist at the lower

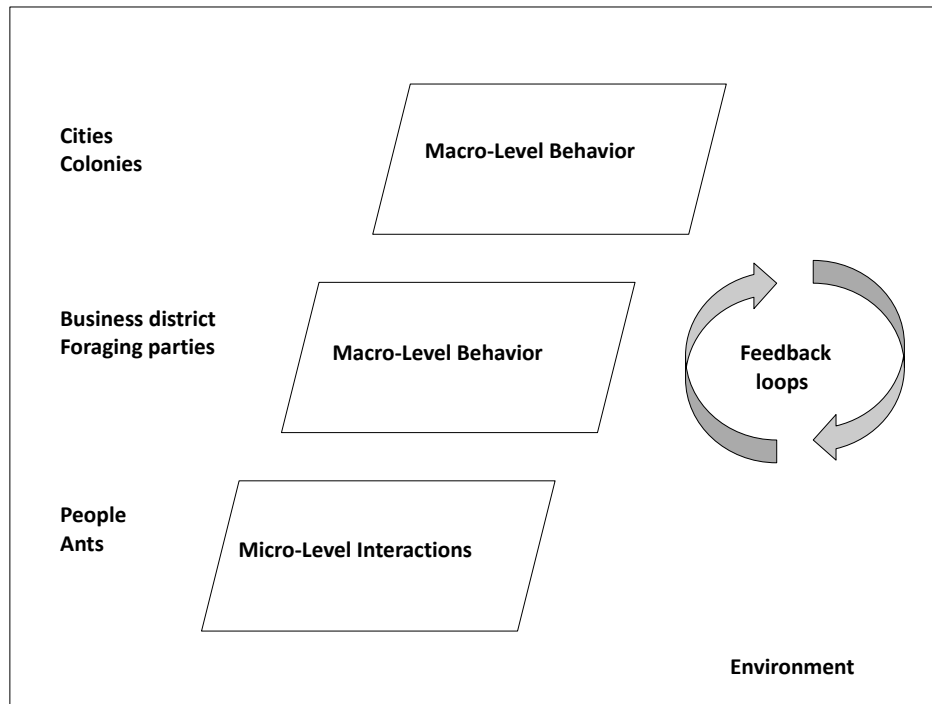


Figure 2. Tiered structure for emergent systems.

levels [112]. For example, a group of business entrepreneurs begin building shops near one another, eventually forming a business district. The district is a complex structure composed of the lower-level businesses. As more districts form, a city begins to emerge. The city is an independent entity, with its own properties, but one that is influenced by the ebb and flow of businesses forming and disappearing. If all the businesses went away, so would the city. However, the city also influences the businesses. As more people move to the city, rental rates may go up, possibly forcing some businesses to close. These closures would then affect the business district and, by extension, the city. The interplay between the levels is represented by feedback loops.

It must be noted that there is an eventual breakdown in reducing macro-level into micro-level components, which, are themselves, composed of other entities. For example, one can derive molecules from atoms, then atoms from neutrons, protons, and electrons, and so on.

At some point, causation fails due to a lack of knowledge. Due to the recursive nature of complex systems where agents form macro-agents which, in turn, can form an agent that can then form a macro-agent, and so on (Figure 3), this research starts with defined agents, and works with the hierarchical levels above it, viewing the affects from atoms, molecules or system subcomponent interactions upon the agent as negligible.

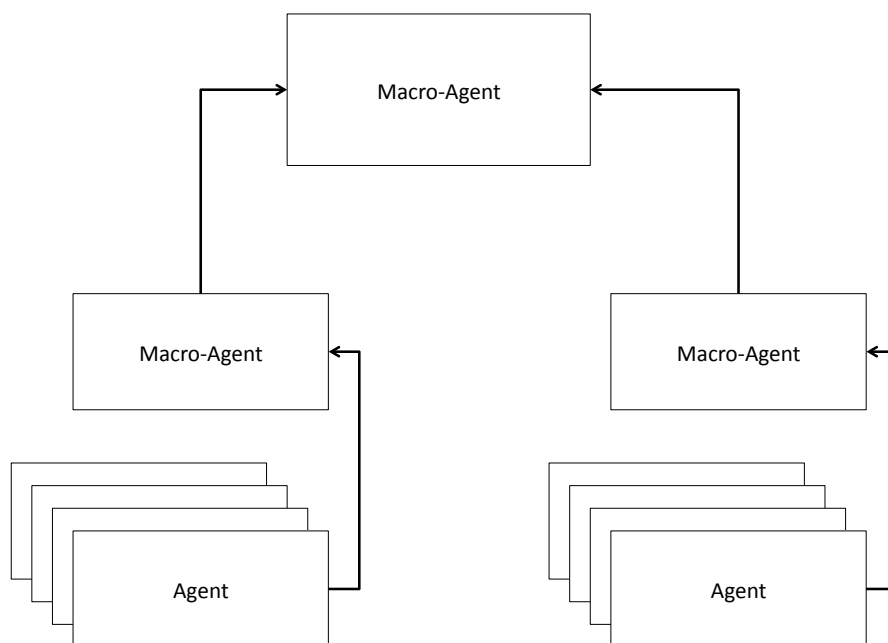


Figure 3. Recursive Agent Formation.

The term *behavior* regularly appears in emergence literature. This work defines a behavior as a culmination of actions. The choices an agent can execute comprise its available actions. Millions of ants foraging for food is a behavior. An individual ant following a pheromone trail to retrieve food is an action. Scout bees signaling the location of a nesting site is an action [103]. Bees collectively choosing and then swarming to the potential nest site is a behavior. In this manner, interactions between agents is limited by the actions those agents can execute and, when aggregated, form a behavior.

This work proposes that agent speciation moves agents towards functional niches which begins building a hierarchical system. Therefore, agent interactions are a result of morphological or physiological influences. One way for agents to interact with each other is through communication.

Agents can communicate either directly or indirectly. *Stigmergy* [70] is indirect communication between agents accomplished by one agent perturbing the environment in a manner that will impact another agent's actions. An ant's pheromone trail is an example of stigmergy. When an ant encounters another ant's pheromone trail, if the scent is concentrated enough, the ant will follow the trail to a food source or even back to the nest. In this manner, one ant affected the action of another without direct communication. Engineered systems, such as a flight of unmanned aerial vehicles (UAVs), use direct communication, i.e. messages, to relay positions, altitudes, etc. A system can use one, or both forms of communication. The important idea to note is that communication between agents is an interaction, and those interactions influence other agent decisions.

As the environment plays a crucial role in the development and direction of a complex system [27, 57, 99], any study of emergence must include it as part of the system structure. The environment surrounds the complex system and provides constant feedback, influencing agent actions, and providing a communication medium. For natural systems, the environment can extend towards infinity which is not overly helpful. Holland [57] proposes the word *niche* to describe a finite boundary for a complex system. For example, the ant colony's *niche* may be one's front yard, or a field. The *niche* bounds the actions of the overall system. Ecologically speaking, *niche* refers to the role or position of an organism in its community [1]. In this work, *niche* describes an agent's community, which is based on the agent's current *function*. For example, a group of foraging ants creates one *niche*, with a group of brood-sorting ants creating another. In this manner, bounded structures appear that possess agents, signals, and emergent behaviors, providing a leverage point for

analyzing their interactions with one another and the system levels above them.

Finally, a complex system must adapt to changing environments [70]. Agent speciation provides a mechanism for system adaptation. For an ant colony, the switching of function – from foraging to nesting to nest defense – provides adaptability in the face of a changing environment [13]. Individual agents, dispersed across a system hierarchy and guided by particular roles, execute actions that aggregate to a global goal. Arguably, engineered agents can learn to maximize a reward signal from the environment to simulate the adaptability of natural complex systems [106]. This creates a robust and adaptable complex system.

This work uses these ideas to formulate the speciation entropy algorithm (SEA) for developing multi-agent systems based on speciation. Different *species* of agents create functionally aligned *niches* which aggregate into macro-level behaviors, eventually culminating into a system-level goal. These niches form the first hierarchal level of the system and align to sub-goals, forming a secondary level. These sub-goals then align to either other sub-goals or system goals, depending on how one split the system-level goal apart during system design, forming tertiary, quaternary, etc., levels. Simple agent speciation - morphological or physiological - allows agents to adapt based upon the feedback provided by higher system levels and the environment - macro to micro influence. Adaptation could be an actual internal change, i.e. internal thresholds for choosing a task may change, or the external switching of functional niches. These changes impact each level of the system - micro to macro influence. In this manner, one instantiates theoretical emergence concepts into engineered systems.

2.4 Entropy

This section discusses entropy and entropy-based research starting with Shannon’s entropy definition from 1948. From Shannon’s entropy, the section presents current research

centered on using entropy based measures to predict agent decisions as well as influence agent decisions. The ideas covered in this section underpin the entropy based system order metric proposed in Chapter IV as well as providing agent decision guidance in Chapters V and VI.

Entropy as a measure of information began with Shannon in 1948. Borrowing from the thermodynamic view of entropy [46], Shannon created a formula measuring the amount of information one could extract from the probability of the occurrence of an event correlated to the observance of such an event [118]. As entropy is a measure of order, or disorder, in a system [46], then using Shannon's equation (Equation 1), one can expect the entropy of a system to decline as an observer gains information about the system under study. The enticement for using entropy measurements with relation to emergence is clear: if one can tie emergent behaviors to the rise, or fall, of system entropy, one builds a possible mechanism for emergent behavior detection.

Upon initial observation, it seems one can correlate a decrease in entropy with information gain in the system; possibly the emergence of some type of pattern that places order upon the system. The problem with using system entropy as an indicator of order is innocently hidden inside the p_i term which represents the probability of finding the system in state i [46]. Calculating every possible state for a system is usually not practical.

$$H = - \sum_i p_i \log_2 p_i \quad (1)$$

However, Mnif and Müller-Schloer [94] propose discretizing the system states by focusing on an attribute of interest. In this manner, one reduces the number of system states. It also adds some meaning to the entropy measure as one now correlates an observed behavior to a reduction in system entropy. The key here is coupling the micro and macro level entropy correctly [105].

Recently, three major works have argued that entropy measures can either indicate the current state of the system (e.g. its level of self-organization or emergence), or drive agent actions towards global intelligent behaviors. Parunak and Bruecker [105] showed that directional entropy, a pheromone guidance approach for a walker searching for a target in unknown space, increases as walkers randomly move about, but reduces as agent knowledge (i.e. a walker encounters pheromones guiding it towards a target) increases, showing that information gain drives down the directional entropy of the system. Additionally, this coupling of directional entropy to agent actions created a metric for measuring system state.

Wissner-Gross and Freer [141] and Mann and Garnett [90] took a completely different approach. Instead of measuring micro-level entropy, they utilized macro-level entropy scores to guide micro-level agent decisions in complex systems. Wissner-Gross argued that causal entropic forces, based on maximizing system entropy, guide agents towards behaviors known as group intelligence (i.e. self-organization). In their view, agents independently choose actions that move a system towards macro-states with the highest causal entropy [141].

Mann and Garnett [90] extended Wissner-Gross and Freer's work showing that multiple agents choose paths based on the path's entropy measure. They argued that groups of agents will select a path that results in more future options, as those options are unknown, then agents select the path with the highest entropy with respect to the number of future states. By viewing future paths as independent Galton-Watson processes, they could assign extinction possibilities to future states, thus limiting the number of future possible states in unknown environments. The problem with their approach, is one must guess the type of distribution that future states will possess. Both [90] and [141] suffer from dimensionality issues as examples of how agents choose paths in future states were either very limited, or left to theoretical bounds.

Another issue with [141]'s work, and by extension [90], is that their agents seem to go

against known thermodynamic laws. For example, [141] released an agent into a bounded square, the agent settled in the middle as it gave the agent the maximum number of future options available through Brownian motion. This goes against basic physical principles where the agent would most likely bounce around the enclosed space, or remain stationary if held to zero energy. If used as a real-world model, it too fails as animals placed into an enclosure will normally move to a corner out of survival instinct – keep a watch out for both predator and prey. Both also fail to adequately explain how to deal with an almost infinite number of possible future macro-states. With this in mind, this work follows [105], where entropy increase and decrease at the micro-level influences macro-level behaviors and that, in the natural world, survival drives a natural system through cycles of maximum and minimum entropy dependent on the tasks at hand.

The heavily studied ant colony provides an example. As the colony's ultimate goal is survival, it depends on thousands of individual ants to complete multiple daily tasks (e.g. foraging, nest clearing, brood sorting, defense, etc.). Through evolutionary means, the simple ant developed an efficient way to deal with the daily environment which is necessarily different than the day prior. One can view the environment as one having maximum entropy with respect to resource locations. The ant does not have a long memory nor does the colony have a map of the local area. Each day, hundreds of ants must leave the nest and randomly search the environment. The random search pattern is accompanied by a signaling pheromone that other ants sense as they walk about the environment. When food is located, the scent changes. The ant lays a new pheromone indicating it is carrying food back to the nest. Other ants will then begin following the trail to the food and back home again. In this manner, ants efficiently find and gather food.

From an entropy perspective, once food is located, the entropy of the colony with respect to resource location declines. As more ants engage in the task of resource collection, entropy continues to drop. However, an interesting phenomena begins to occur: as the num-

ber of ants engaged in resource collection grows, the likelihood of another ant beginning to collect food decreases. Somewhere, in the primitive ant mind, a critical threshold turns the ant away from food gathering and towards other tasks. These behaviors allows ants to complete multiple tasks concurrently. These behaviors have a common theme: sharing of information. The more efficiently information spreads throughout the system, the more tasks the colony can accomplish. The resultant multi-task emergent behavior is based on ants maximizing their local entropy with respect to internal bias, i.e. physiological predisposition.

The proposed hypothesis is that system level order emerges as agents align into functional niches based primarily on their physiology. If one assumes that agents wish to move towards higher, entropic states [90, 105, 141], then one could use entropy as an agent guidance method. In this manner, entropy serves as an internal, physiological component that influences agent decisions. Chapters IV, V, and VI explore this concept as agents use local stimuli from the environment and other agents, plus entropy calculations, to choose actions and self-organize into functional niches. Results from these chapters show that entropy-based approaches create robust and adaptable, multi-agent systems.

2.5 Genetic Algorithms

Genetic algorithms (GAs) are a search and optimization technique based on natural selection and genetics [55] and fall under the area of evolutionary computation [70]. The chromosome for a GA based agent represents a possible solution to a problem and can consist of agent actions or capabilities (Figure 4). At the beginning, the algorithm constructs a population of random chromosomes. It then applies these possible solutions the problem domain and assigns a value to each via a fitness function. The population is then sorted and the algorithm selects the fittest individuals to create the next generation. A new chromosome is created by copying parts of each parent chromosome. A swap between

parent chromosomes is performed during the copying process at a predefined crossover point. Finally, a mutation function iterates over the chromosome, changing values with a low probability. This function helps GAs escape local minima, or maxima, as it forces the chromosome to explore other solutions. Eventually, the population will converge to a solution. As GAs can run for an infinite amount of time, usually one sets a limit on the number of generations to explore before exiting the algorithm.

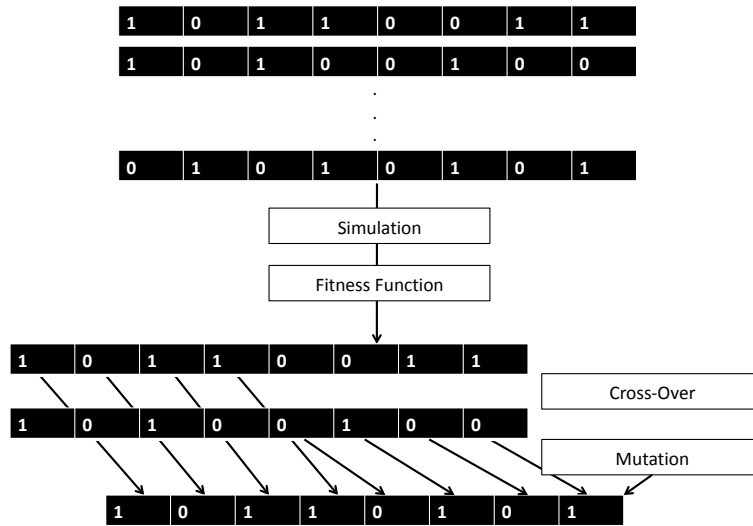


Figure 4. Example genetic algorithm. An initial set of chromosomes (top of figure) run through a simulated environment. Afterwards, a fitness function assesses each chromosome's performance, selecting the best scoring individual's for breeding. Breeding is accomplished via the cross-over operation. Finally, each part of the chromosome is given a small probability of mutating into another form (e.g. switch from 0 to a 1). The process starts again with the new population and continues until encountering a pre-determined stopping condition.

Researchers have successfully used GAs on optimization problems ranging from circuit layouts and job scheduling [114], to flight planning and cooperation [48, 88, 86], to autonomous decision making in aerial combat situations [50, 31, 36, 37]. Two factors contribute to the appeal of GAs. One, the idea of evolution via natural selection is intuitive and one could say its attractiveness lies in the understanding of its role in evolution in the world that surrounds us. Two, GAs find solutions to NP-hard problems such as the Traveling

Salesman Problem (TSP) [45, 100] and the NP-complete knapsack-problem [22, 71].

While many GA applications focus on the fitness of individual solutions, Tarapore [125] and Duarte [33] assessed the fitness of entire populations. Both studied the effects of changing agent structure, i.e. adaptive response thresholds, on division of labor in ant-like populations. Each ran simulations where multiple colonies were assessed via a global fitness function where sub-populations of the most successive colonies were used to create the next set of colonies. Chapter VI uses the same methodology to test how agent decision functions, resource constraints, and the environment, affect division of labor in ant colonies. Results show that both the choice of agent decision function and environmental conditions have a larger impact on colony performance than resource constraints. Furthermore, simulation data indicates that colonies with lower speciation do not outperform those with higher speciation, which goes against current thoughts on the role of high specialization, i.e. low speciation, and colony resource output [33].

This chapter covered the major topics used by the rest of this work to answer the research questions posed in Chapter I. Thematically, emergence research and entropy-based decision functions are found throughout the following chapters. Emergence helps frame the work and provides theoretical guidance on what system properties could lead to desired macro-level behaviors for Chapters III - VIII. Entropy plays a role in both guiding agent decision in Chapters IV, V, VI and VIII. Finally, GA provide an evolutionary component for agents in Chapters VI and VIII, enabling the development of agent species *in real time* as the system executes.

III. Informal Team Assignment Algorithm

3.1 Introduction

The second and third research question ask how agent speciation impacts system robustness while creating multi-task capabilities.¹ To answer these questions, this chapter 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 multiple tasks. The coupling of predefined roles and actions with inter-agent signaling, allows the swarm to create multiple niches with agents dedicated to common tasks such as patrol, circle and pursuit.

These roles represent an agent’s physiology. Agents functionally align to one of their respective roles at each time step dependent upon stimuli from the environment and other agents. These self-organized groups, labeled niches, create an emergent system. The ITAA provides a scalable, control-solution for UAV swarms and provides those swarms with broad task capabilities based upon agent speciation. Simulations support the overall hypothesis that agent speciation creates robust and adaptable systems.

Simulation data also shows that even strictly controlled and defined systems can create emergent phenomena. In this work, two interesting behaviors emerged. One, agent speciation and signaling led to the formation of small teams of pursuing agents that invoked temporary leadership behavior noted in pigeon flocks [21]. This is a notable finding, as the approach created collaborative teams without predefined leader-follower roles or a centralized control authority. Two, small teams of pursuit agents self-built tactical nets around invading agents, effectively surrounding them, increasing the odds of capture. Both of these emergent patterns emerged without predefined coding heuristics or definitions.

The rest of the chapter is organized as follows. Section 3.2 reviews current autonomous

¹The majority of this chapter was published in the proceedings of the Florida AI Research Society (FLAIRS-31) Conference held in Melbourne, FL, May 2018.

swarm control research. Section 3.3 formally presents the algorithm. Section 3.4 describes the experimental variables and scenarios, with Section 3.8 providing data analysis and discussion. Section 3.9 concludes the chapter.

3.2 Background

Finding methods for decentralized control of autonomous swarms is an active area of research [91, 17, 129]. Tan, et al. [123] 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 swarm to alter its structure through collaborative team formation, making the swarm robust to dynamic changes in the environment.

Tolba, et al. [129] 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 [110] 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. [31] 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 commu-

nicate 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. [8], 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 [129], 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. [2] 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 [2], et al., while including agent signaling found in [129] and [8]. 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.

3.3 Algorithm

Emergence research [57] provided the inspiration for the ITAA. Emergence theorizes that the aggregated interactions between simple agents give rise to complex behaviors [10, 55, 93, 117]. Holland [57] 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. [13] 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, $R = \{r_1, r_2, \dots, r_n\}$. Each role contains a threshold setting, $T = \{t_1, t_2, \dots, t_n\}$, which is composed of coded logic that determines when an agent changes its role. Finally, one assigns a set of actions, $A = \{a_1, a_2, \dots, a_j\}$, to each role.

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

```

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).

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.

3.4 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 [2]. However, this work differs in two ways from other UAV based PEG scenarios [2, 85, 133] in that (1) large pursuit teams (up to 20 agents) were tested and (2) evading agents possessed superior speed.

3.5 Scenario

Figure 5 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.

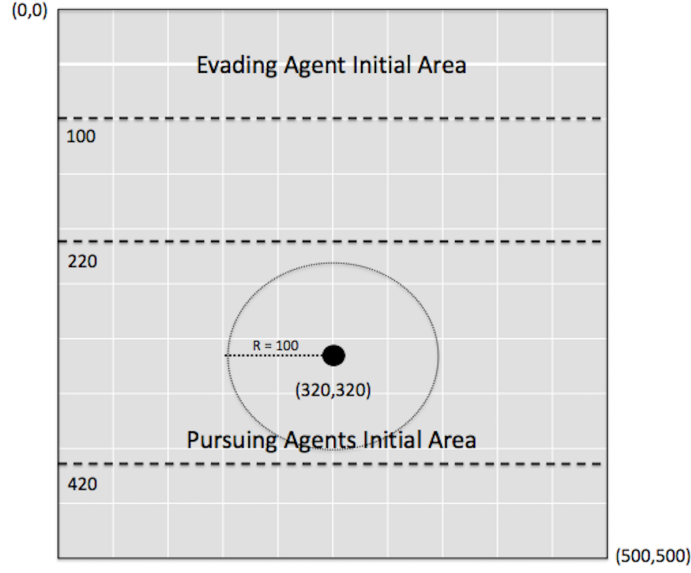


Figure 5. Grid layout of simulation environment. Both initial position boundary rectangles for evading and pursuing agents are shown. The middle ellipse portrays the orbital radius followed by agents invoking the *circle* role.

3.6 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 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.

Algorithm 2 : Nest Defense

```
1: procedure ROLEASSIGNMENT
2:   changeRole(environmentState)
3:   executeActions( $r_{current}$ )
4: end procedure
5:
6: procedure CHANGEROLE(STATE)
7:   for  $r \in allRoles$  do
8:     thresholdMet( $r$ , STATE)
9:   end for
10: end procedure
11:
12: procedure THRESHOLDMET(ROLE, STATE)
13:   if detectEnemy(state) or signal received then
14:      $r_{current} \leftarrow pursuit$  return
15:   else if ( $state.distance.nest > 100$ ) then
16:      $r_{current} \leftarrow circle$  return
17:   else
18:      $r_{current} \leftarrow patrol$  return
19:   end if
20: end procedure
21:
22: procedure DETECTENEMY(state)
23:  $allObjectsInRange \leftarrow object.state.distance \leq 100$ 
24:   for  $allObjectsInRange$  do
25:     if  $object == enemy$  then return true
26:   end if
27: end for
28:   return false
29: end procedure
30:
31: procedure EXECUTEACTIONS(ROLE)
32:   if  $role == pursuit$  then
33:     signalNeighbor()
34:     seek evading agent
35:   else if ( $role == circle$ ) then
36:     orbit + separate
37:   else
38:     separate
39:   end if
40: end procedure
41:
42: procedure SIGNALNEIGHBOR
43:   for  $neighbors \in distance$  do
44:     signal
45:   end for
46: end procedure
```

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.

3.7 Control and Independent Variables

Table 1 presents the experiments' control and independent variables. Simulations changed evading agent speeds, pursuing agent team sizes, and signaling actions. Figure 6 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.

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

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

ROLES

		Patrol & Pursuit	Patrol, Circle, Pursuit
S I G N A L	True	Evading Speed: 0.50, 0.75, 1.00, 1.25, 1.50	Evading Speed: 0.50, 0.75, 1.00, 1.25, 1.50
		Pursuing Agent Swarm Size: 5, 10, 15, 20	Pursuing Agent Swarm Size: 5, 10, 15, 20
	False	Evading Speed: 0.50, 0.75, 1.00, 1.25, 1.50	Evading Speed: 0.50, 0.75, 1.00, 1.25, 1.50
		Pursuing Agent Swarm Size: 5, 10, 15, 20	Pursuing Agent Swarm Size: 5, 10, 15, 20

Figure 6. Experiments

3.8 Results

Tables 2 and 3 summarize pursuing agent win ratios as both the number of pursuing agents and evading agent speed increased. Table 2 holds the baseline results as agent roles were limited to patrol and pursuit (patrol-pursuit) with signaling enabled. Table 3 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 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.

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

		Number of Pursuing Agents			
Evader Speed		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 3. Circle-Pursuit agent win ratios versus evading agent speed.

		Number of Pursuing Agents			
Evader Speed		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

To test the impact of signaling on swarm performance, signaling actions were removed from both patrol-pursuit and circle-pursuit swarms for comparison (Figures 7 - 10). 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.

The circle-pursuit teams fared better against evading agents across the majority of evad-

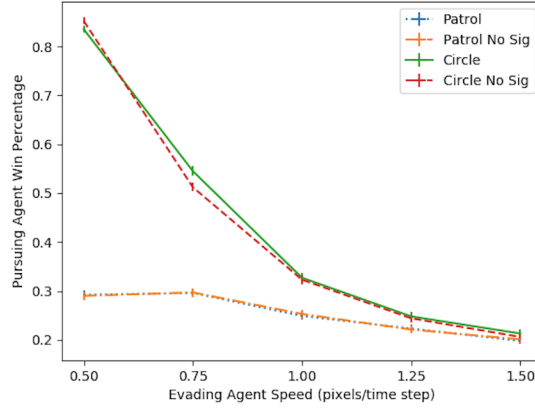


Figure 7. Win percentages for 5 pursuing agents pitted against a single evading agent.

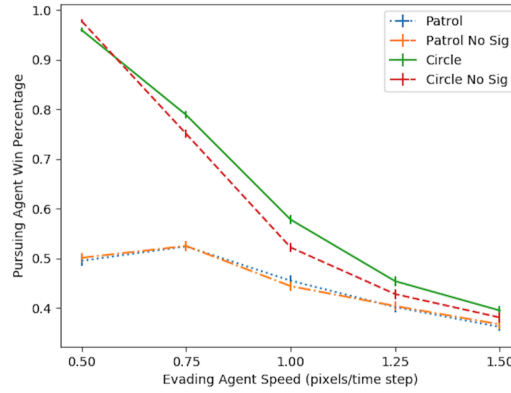


Figure 8. Win percentages for 10 pursuing agents pitted against a single evading agent.

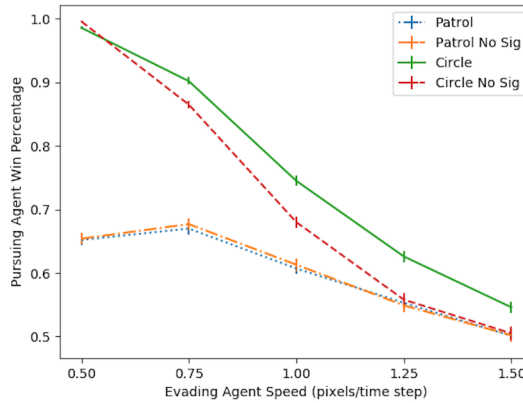


Figure 9. Win percentages for 15 pursuing agents pitted against a single evading agent.

ing 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

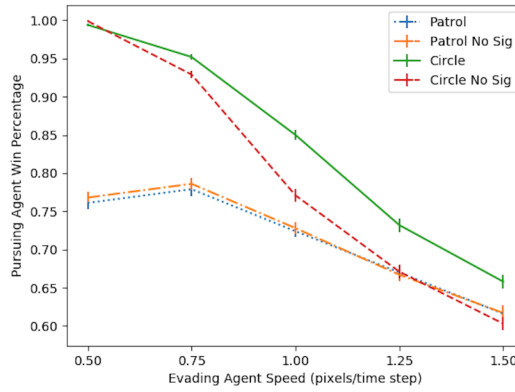


Figure 10. Win percentages for 20 pursuing agents pitted against a single evading agent.

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 [21]. 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 42 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 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 12 provides the win rates for defending teams of 20 agents as they were the most successful

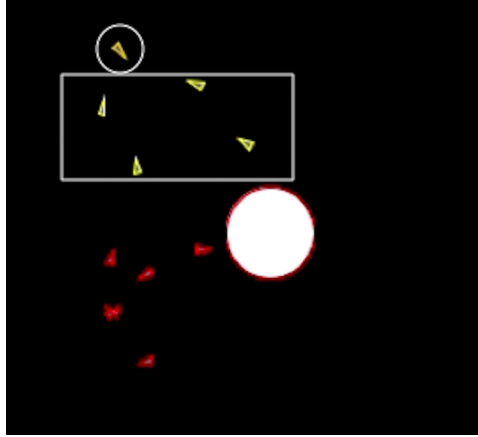


Figure 11. During this simulation, four pursuing agents in *pursuit* roles (boxed) converge on an evading agent (circled). Other pursuing agents remain in their current roles (*circle* or *patrol*). The white dot represents the nest.

in the single evader trials.

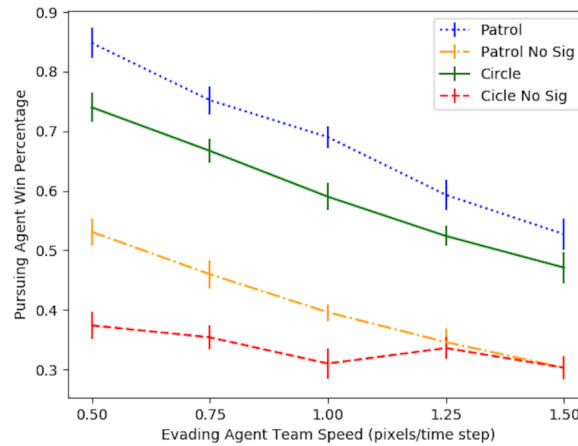


Figure 12. Win percentages for 20 pursuing agents pitted against three evading agents.

Surprisingly, in this scenario, patrol-pursuit teams with signaling fared 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.

3.9 Conclusion

This chapter 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.

The dynamic, self-organized structuring of agents into functional niches, i.e. groups of agents engaged in the same task, led to multi-task accomplishment inside the UAV swarm, supporting this work's primary hypothesis. This chapter also highlighted the effectiveness of decentralized control approaches for controlling an UAV swarm. These two system traits led to the emergence of dynamic teams capable of detecting and intercepting a superior opponent.

The results presented here show that decentralized control algorithms can be built in a general manner where the addition of new capabilities is relatively easy. The modular design, i.e. different physiological traits, avoids large code rewrites as only new behaviors and the roles that associate to them, need to be coded into the system. Additionally, the method allows full autonomy. The agents work free of centralized control and towards a common goal. This work also highlights the importance of communication, or *signals* [57], for purposeful emergent behaviors to appear.

The second and third research questions asks how agent speciation impacts system robustness as well as create multi-task capabilities. The ITAA presented in this chapter built upon agent speciation, described as agent roles, to achieve an adaptable system that could accomplish two tasks at once: detection and capture. The desired macro-level be-

havior, i.e. invading agent capture, was achieved by the UAV swarm as agents dynamically switched roles as the environment changed, i.e. invading agents were detected. This behavior indicates an adaptable system. In the three evading agent environments, the system showed robustness as the swarm self-organized team responses to numerous invaders at once. Although the swarm split into smaller sub-swarms, the system was able to complete its macro-level goal of intercept, showing robustness in a dynamic environment.

IV. A Macro-Level Order Metric for Complex Systems

4.1 Introduction

This chapter proposes a metric that can measure how speciation leads to robust, hierarchical, self-organized systems.¹ Agents presented in this chapter use both speciation and Shannon’s entropy to achieve macro-level goals, exploring research questions one and three. Results show that speciated populations, using entropy to guide agent decisions, create hierarchical self-organized systems with multi-task capabilities.

This chapter focuses on identifying and leveraging emergent, or self-organizing, behaviors in a complex adaptive system. One of the major research problems associated with complex systems is the credit assignment problem [56] because agent performance (if using a reward based fitness function) relies on the actions of other agents. The payoff is located at a global level and may appear much later in the system’s performance [56]. However, global knowledge or temporal considerations for positive agent actions may be unnecessary. For one, in most natural complex systems, agents do not possess global knowledge nor do they have long term memories to correlate action-effects [107, 109]. Instead, agents base decisions solely on the local information they currently possess.

Analyzing how agent interactions affect macro-level self-organized behaviors can yield a deeper understanding of how complex adaptive systems work. The dynamic nature of complex systems makes it difficult to determine if, or when, a system has reached a state of equilibrium or is about to undergo a major transition reflecting the appearance of self-organized states. Using the notion of local neighborhood entropy, this chapter presents a metric for evaluating the macro-level order of a system. The metric is tested in two dissimilar complex adaptive systems with self-organizing properties: an autonomous swarm searching for multiple dynamic targets and Conway’s Game of Life. In both domains,

¹The majority of this chapter was published in the proceedings of the 12th IEEE International Conference on Self-Adaptive and Self-Organizing Systems (SASO 2018), held in Trento, Italy, in June 2018.

the proposed metric is able to graphically capture periods of increasing and decreasing self-organization (i.e. changes in macro-level order), equilibrium and points of criticality; displaying its general applicability in identifying these behaviors in complex adaptive systems.

This chapter explores the idea of using local agent information, modeled by informational entropy, to guide agent decision in a swarm tasked with covering multiple dynamic targets. Additionally, this work couples local entropy measures to macro-level order to create a metric capable of identifying the emergence of self-organized behaviors in a complex system as well as periods of system stability and points of criticality. Overall, the simulations presented here investigate all three research questions. One, how agent entropy influences agent physiology and agent decisions at each time step, making the swarm robust and adaptable? Two, how does the use of informational entropy affect system outcomes? And, three, if incorporated in a decentralized-manner, does information entropy lead to broad task capabilities?

4.2 Motivation

Complex adaptive systems (CAS) are systems composed of a large population of agents that interact and adapt to their environment [56]. The aggregated interactions of these agents form complex behavior patterns [57] which appear to exceed the summation of those interactions [103]. In other words, the whole is greater than the sum of its parts. Historically, these emergent [55] or self-organizing [15] behaviors make these systems difficult to control [25]. Additionally, CASs pose significant challenges for researchers as they operate in dynamic environments with imperfect and incomplete information [101]. Engineered CASs are being fielded and interconnected every day [25], with this increased deployment comes an increased interest in developing methods for identifying possible self-organizing behaviors in complex systems. As agent interactions compose the driving

force of a complex system's macro-level behavior [108], then tying those interactions to macro-level self-organizing patterns becomes the critical component for behavior analysis.

Whether it is a natural or engineered complex system, periods of stability and points of criticality occur during the system's lifetime. Points of criticality indicate a transition of the current system state to another, which may be a stable, unstable, or self-organized state [12]. For example, the collective rush of ants to gather a new food source will dominate the colony's short-term behavior. However, as the food dissipates, ants begin reverting back to other tasks such as brood sorting, nest defense and refuse cleaning; ensuring the colony's survival [137, 83]. Natural systems undergo periods of stability, e.g. a fairly even distribution of labor across all known tasks, and critical events, e.g. discovery of food creates an imbalance in ant task distribution. Langton [81] showed that periods of quiescence, as well as chaos, occur in cellular automata (CA), demonstrating that engineered CASs contain similar patterns to those found in natural systems. Cycles of stability in a CAS are indicative of a robust system as it is able to adapt to changes in the environment. Although prediction of such transitions and cycles is nearly impossible in real-time systems engaged in dynamic environments, it is possible to detect periods of stability and critical points in an *ad hoc* manner.

Inspired by entropy based approaches found in [116, 105] and newer research by [141, 90], this paper presents a macro-level metric for identifying periods of increasing and decreasing order. Order is the appearance and disappearance of self-organized behavior and indicates the stability and points of criticality in self-organizing systems. By coupling local entropy scores with agent distribution across neighborhoods, the metric captures macro-level agent dynamics, enabling detection of system stability and points of criticality. The metric is applied to both an autonomous Unmanned Aerial Vehicle (UAV) swarm engaged in a dynamic multi-target surveillance scenario and Conway's Game of Life. In both domains, the establishment of neighborhoods and population growth affect macro-level order.

If neighborhoods remain static, i.e. the loss of population is minimal, the order of the system remains stable. When communities die off, or appear, noticeable departures in system order occur. Results show that the proposed metric is able to detect changes in system structure due to self-organized behaviors.

4.3 Related Work

Schulman and Seiden [116] first broached the idea of using system entropy, based on Shannon’s entropy [118], as an indicator of order for Conway’s Game of Life in 1977. More recently, Parunak and Bruecker [105] showed that directional entropy, a pheromone guidance approach for a walker searching for a target in unknown space, increases as walkers randomly move about, but reduces as agent knowledge (i.e. a walker encounters pheromones guiding it towards a target) increases, showing that information gain drives down the directional entropy of the system.

Wissner-Gross and Freer [141] and Mann and Garnett [90] take a slightly different approach. Instead of measuring micro-level entropy, they utilize macro-level entropy scores to guide micro-level agent decisions in complex systems. Wissner-Gross argue that causal entropic forces, based on maximizing system entropy, guide agents towards behaviors known as group intelligence (i.e. self-organization). In their view, agents independently choose actions that move a system towards macro-states with the highest causal entropy [141]. Their algorithm partitions the current system state into a set of possible future states and using Equation 2, calculates the entropic force (F) for that state.

$$F(X_0) = T \nabla_x S(X) \quad (2)$$

Here, T represents the reservoir temperature, $S(X)$ is the entropy associated with the macro-state X , and X_0 , is the current macro-state. In this manner, one can evaluate multiple paths

through possible future macro-states with agents choosing paths with the highest causal entropy. The main premise is that organized intelligence is due to individual agent decisions that result in future states that possess the most options for the agent (i.e. highest causal entropy). The paradox being that the organized behavior of these agents emerges although they collectively choose to increase the macro-level entropy of the system.

Mann and Garnett [90] extended Wisser-Gross and Freer’s work by viewing future paths as independent Galton-Watson processes. This allows them to assign an exponential distribution to unknown future states, conditional on path extinction probabilities [90]. Again, consensus decision making (i.e. self-organized behavior), is the result of agents selecting maximum causal entropic paths. However, both approaches assume that: a) agents somehow know how to increase macro-level entropy and, b) possible future paths, and states associated with them, are countable, which is one of the difficulties in trying to use entropy measures in complex environments where state search spaces grow exponentially.

Shannon’s definition of entropy [118] illuminates why entropy measures are interesting. Intuitively, if the amount of information discernible from a system grows, the system’s macro-level entropy decreases. The correlation between a decrease in system entropy with the emergence of a pattern (i.e. order implies an information gain) makes Shannon’s entropy equation a tempting approach for detecting self-organizing or emergent behaviors in a complex system. However, p_i in Shannon’s equation (Equation 3) represents the probability of finding the system in state i [46]. Calculating every possible state for a system, especially in continuous environments, can quickly become computationally infeasible. It becomes even less practical in a real-time system where agents make decisions in timescales as small as microseconds.

$$H = - \sum_i p_i \log_2 p_i \quad (3)$$

One way to overcome this limitation is to discretize system spaces to reduce the number of possible system states. The benefit of such an approach is it allows one to expediently calculate Shannon's equation to measure system entropy. For example, although working in continuous space, Parunak and Brueckner [105] placed a grid over the system space, subsuming multiple states into separate partitions, allowing one to count the number of agents in those partitions which then define the system state. Mnif and Müller-Schloer [94] generalized the approach where one selects an attribute of the system that is discrete and enumerable. By doing so, one focuses on attributes of interest in which entropy measures carry more precise meaning while reducing the number of system states, making the use of entropy measures tractable. However, as [105] points out, one must couple micro and macro-level entropy in order to capture the imposition of order upon the system.

Parunak and Brueckener's idea is modeled after emergence researchers such as Holland [55] who argue that macro-level patterns rely on changing micro-patterns and it is through the aggregation of simple agent interactions that higher-level emergent behaviors arise [108]. What distinguishes emergent behavior at the macro-level is the absence of those behaviors at the micro-level [112]. The agents at the micro-level contain no knowledge beyond their immediate environment, or 'niche' [57]; however, their aggregated actions create complex patterns in their *niche* which then aggregate to create complexity in the ecosystems that subsume them. Agent knowledge is key here, unlike [141] and [90], global knowledge for agents is not realistic. Self-organized, or emergent patterns, develop from limited local agent knowledge. The autonomous swarm introduced in Section 4.4 will show how agents with limited knowledge can create recognizable, global patterns by choosing actions leading to the highest local entropy state. As macro-level entropy changes may indicate some type of order change or self-organized behavior [15], one must consider the accumulation of local entropy at the system level. However, mere averaging or summation of local entropy fails to convey much meaning about the type of order or behavior that

is occurring and, in some cases, could fail to detect self-organized behavior [12]. In order to capture more information about agent interactions, this work proposes to tie both micro-level entropy and macro-level population density together to achieve a metric of system order that can also detect periods of stability and points of criticality.

The metric is based on Bonabeau, et al.’s threshold function (Equation 4), introduced as a model of task division in insect societies [13]:

$$T_{\theta}(S) = \frac{s^n}{s^n + \theta^n} \quad (4)$$

In their equation, the probability of performing a task is based on a response threshold, θ , related to a stimulus, s , associated with a task, and, n , determines the steepness of the threshold. Accordingly, if $s \ll \theta$, the probability of performing the task is almost 0, while if $s \gg \theta$, then it is close to one [13]. This type of interplay between the stimulus, s , and threshold, θ , is exactly the type of behavior one needs to capture between micro and macro-level behaviors. Section 4.4 shows an extension that provides a metric of macro-level order with enough sensitivity to produce a reasonable measure of system order that indicates both times of periodic stability and points of criticality. As the metric is tied to population density, system order also provides general information about the number of neighborhoods and the distribution of the agent population across them.

Using Mnif and Müller-Schloer’s [94] generalization approach, system states are measured as the number of agents assigned to specific neighborhoods, making state probability calculations tractable, enabling the use of entropy measurement. For this research, agents in the UAV domain continuously seek to maximize the entropy value of their local neighborhood. The constant pursuit of entropy maximization results in self-organized teams with shared common tasks. In the Game of Life, local entropy values are evaluated but do not influence agent decisions. However, coupling micro-level entropy and population

distributions to macro-level order enables the identification of phase transitions (e.g. group emergence, self-organization) in the system. In both domains, local entropy scores and population distributions across neighborhoods produce an order measurement that is sensitive to micro-level dynamics.

4.4 Approach / Methodology

This work presents a metric that uses local entropy measures as a part of calculating macro-level order. As domains change, local entropy measures necessarily require different calculations. However, the macro-level order metric only needs the local entropy values. In other words, at the macro-level, the metric relies on one's definition and implementation of local entropy. The macro-level order metric is defined by:

$$\tau_{macro} = \frac{1}{|N|} \sum_{n=1}^{|N|} \frac{S(n)^2}{S(n)^2 + \Delta_n^2} \quad (5)$$

N is the set of all neighborhoods, $\{n_1, n_2, \dots, n_{|N|}\}$, present in the system. $S(n)$ is the local entropy associated with neighborhood, n , and Δ_n is the number of agents in that neighborhood divided by the total number of agents. Macro-level order is measured by a summed average of local neighborhood entropies, $S(n)$, and the macro-level distribution of agents, Δ_n , normalized by $|N|$. By using this equation, macro-level measurements carry indications of what is happening at the micro-level. For example, if the number of agents associated with a neighborhood, Δ_n , increases and begins surpassing $S(n)$, macro-level order decreases. This indicates an imbalance of agent distribution. For example, a larger percentage of the agent population lives in one area. If the sum of the local entropies of all agents begins to dominate, Δ_n , then macro-level order increases towards 1. Higher values indicate the emergence of multiple self-organized neighborhoods. Periods of equilibrium occur when the agent population is fairly evenly distributed (i.e. $S(n) \approx \Delta_n$) and remains

relatively constant.

As recommended by a critique on the application of metrics across domains [12], the proposed macro-level order metric is tested in two dynamic domains: a UAV swarm engaged in a multi-task environment and the Game of Life.

The first domain of study consists of a swarm of UAVs engaged in intelligence, surveillance and reconnaissance (ISR) missions. When the simulation starts, all UAVs fly randomly through a bounded grid (1,000 x 1,000 pixels). Random targets, modeled as tasks, are added to the simulation at random locations and times. Any target (i.e. task) within range of a UAV sensor is added as a task to its task list. Agents calculate their local entropy by polling neighboring agents to find the distribution of agents across known targets. The agent proceeds to add itself to the number of agents assigned to task 1, and calculates its entropy using Equation 6 but instead of A_T equalling the number of agents in the entire simulation, it equals the number of agents in the agent's neighborhood. It continues the calculation for each known task, selecting the target that yields the highest local entropy.

Targets possess a limited, random, lifespan and disappear once the time limit is exceeded. Once a target disappears, agents must decide on a new target to fly towards, or revert back to random flight if no other targets exists within their sensor range. Fig. 13 and Fig. 14 are screenshots from one simulation run. Agents (triangles) detect three distinct targets (circles). Guided by maximizing local entropy, they self-organize into three teams to cover each target.

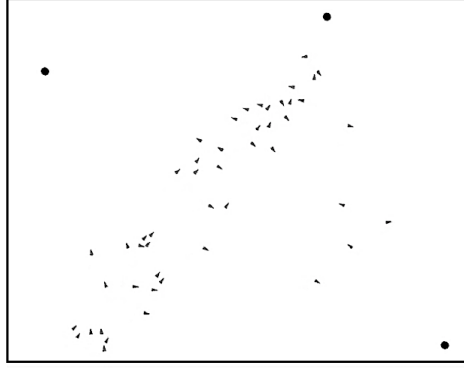


Figure 13. Three targets (circles) detected by the swarm (triangles).



Figure 14. Swarm self-organizes into three teams which begin flying towards selected targets.

The problem facing the agents is multi-fold. One, agents must decide what target, possibly among many, they must engage. Additionally, all decisions are based solely on local information. They can query agents within range about their current actions, but do not have global information on the location of agents outside of communication range or targets outside of sensor range. Finally, target appearances occur in a dynamic fashion. Agents must pursue new targets as they are detected, leading to dynamic task switching and coordination with other agents. However, coordination is limited to known information. Agents do not assume leadership roles nor do they task other agents. Each agent decides their task independently. Also lacking is a central controller. Once the UAV swarm is launched, all agents act autonomously.

For an autonomous UAV swarm in a dynamic, multi-task environment, task entropy is

defined as:

$$S(n)_{uav} = - \sum_i \frac{A_i}{A_T} \log \frac{A_i}{A_T} \quad (6)$$

where A_i equals the number of agents assigned to task i and A_T equals the number of agents in the simulation. Using Equation ??, the macro-level order metric, τ_{macro} , is:

$$\tau_{macro} = \frac{1}{|N|} \sum_i \frac{S(n)_{uav}^2}{S(n)_{uav}^2 + (\frac{A_i}{A_T})^2} \quad (7)$$

For the UAV domain, macro-level order is measured by the normalized sum of task entropies, $S(n)$, the percentage of agents assigned to task i , $\frac{A_i}{A_T}$, divided by the total number of tasks, $|N|$. With these measures, agents switching between tasks create noticeable changes in the macro-level order in both directions. When agents discover new tasks, macro-level order increases, while, as tasks dissipate or agents remove themselves from tasks, macro-level order decreases, yielding indicators of micro-level behaviors at the macro-level. When no tasks are available, macro-level order is simply 0, as agents revert to random search flight patterns.

The second domain is Conway's Game of Life. The basic concept for the Game of Life is a 2D grid where cells are either alive or dead. Depending on the number of neighbors (8 adjacent cells) a cell possesses, then it will either die from overcrowding or isolation. Dead cells may become alive if they possess three neighbors. From these simple rules, complex patterns of oscillation and change occur. Some patterns, such as *gliders*, appear to move across the screen while others, like the Gosper gun, continuously create other structures. For this paper, the environment is bounded by a 100 x 100 grid. This bounding stops gliders from perpetuating forever, resulting in a solid block structure on the edge. What is interesting is trying to identify periods of oscillation, which can indicate periods

of stability, or transformation, as the case with a glider colliding with a wall.

In the Game of Life, local entropy must be calculated differently as agents here do not move anywhere and their decisions are based on a set of rules. Any type of motion associated with the Game of Life is placed upon the system by the observer, making the appearance of ‘movement’ a self-organized behavior that is the result of cells birthing and dying. Schulman and Seiden [116] analyzed various statistical properties of the Game of Life and proposed an entropy measure based on the size of a grain of cells and a proportion of the living cells compared to the expected average (Equation 8). In this manner, they created a probability distribution for the living cells in the game. Instead of setting a set grain size (e.g. 10 x 10 square), the current state of the game is viewed as a graph, G , where every cell is a vertex, V . Each connected subcomponent, δ_s , in G is treated as a grain.

$$S(n)_{life} = -\frac{j^2}{J^2} \sum_1^{|\delta_s|} \left[\frac{A_{ni}}{j^2} \log \frac{A_{ni}}{j^2} + \left(\frac{j^2 - A_{ni}}{j^2} \right) \log \left(\frac{j^2 - A_{ni}}{j^2} \right) \right] \quad (8)$$

Here, A_{ni} is the number of nodes in subcomponent, δ_{si} , j is the total number of alive cells, and J is the size of the grid (e.g. 100 x 100, $J = 100$). The state of the game at time step, t , is treated like a graph, G , where each community is a connected sub-component, δ_s , of G . These calculations feed into the macro-level order metric:

$$\tau_{macro} = \frac{1}{|\delta_s|} \sum_j \frac{S(n)_{life}^2}{S(n)_{life}^2 + \left(\frac{|\delta_{si}|}{j} \right)^2} \quad (9)$$

Instead of dividing neighborhoods up by task, neighborhoods in the Game of Life are divided into connected sub-components, δ_s . As with the UAV domain, changes in the distribution of *live* cells changes the macro-level order of the system. When very few sub-components exists, the system order is low, while increases in sub-components creates

higher macro-level order. The one drawback for the metric with respect to the Game of Life is the appearance of one subcomponent that never changes in size (i.e. population remains constant) but its position changes. For example, a glider moving in continuous space would never change the macro-level order of the system. However, in bounded space, the glider will eventually collide with a wall and transform into a block of four living cells, changing the macro-level order of the system.

4.5 Experimental Analysis

Autonomous UAV Simulation.

The first set of experiments focused on the autonomous UAV swarm utilizing the local maximum entropy decision process. These experiments tested two questions: one, could maximizing local entropy yield complex, macro-level patterns aimed at task accomplishment, and two, does the macro-level order metric indicate periods of system stability and points of criticality. For each scenario, 50 UAV agents were assigned to the available labor pool. The population was homogeneous with each agent possessing the same capabilities, sensor ranges and max fuel capacity. Only initial positions and fuel levels were randomized. In order to refuel, an agent had to disengage from its current task and return to base, creating imbalances in target coverage. Beginning at time step 1,000, up to two targets appear, each at a 45 percent probability.

Scenario 1.

In this scenario, agent sensor and communication ranges covered the entire 1,000 x 1,000 pixel grid, enabling complete knowledge of available tasks as well as allowing agents to coordinate with all other agents in the simulation (see Table 4). Starting at time step 1,000, targets appear and disappear with a 45 percent chance, and at every 500 time steps thereafter. This scenario shows how agents maximizing local entropy creates self-organized

teams and how changes in team membership (i.e. number of agents covering a task) create noticeable macro-level disturbances.

As a plot of labor distribution over time becomes unwieldy beyond one or two tasks, Table 5 contains labor distribution values at possible critical points (i.e. tasks appear or disappear). Fig. 15 plots the macro-level order of the system over time. The initial jump in order occurs when the first two tasks appear, followed by a stable period of task coverage. At 1,500, a drop in order occurs as one task is removed and replaced by another. At 2,000, all tasks disappear from the simulation. At 3,000, a spike occurs due to a balanced distribution of task-assigned and refueling agents, leading to a period of stability, even when a task disappears and is replaced by another at 3,500. This occurs as agents cycle between refueling and encountering tasks afterwards. A noticeable jump occurs at 4,500 when a third task appears and labor distribution remains fairly even.

Table 4. Control and Independent Variables for UAV Scenario 1. All speeds are in pixels per time step, with ranges in pixels. Fuel and life metrics decremented by 1 at each time step.

Control		Independent	
Refueling Center	(500, 950)	Fuel Level	[3,000, 8,000]
Visual Radius	360 degrees	Sensor Range	1,000
Velocity	0.50	Comm Range	1,000
Swarm Size	50	Target Life	[500, INF]

Table 5. Number of Agents per Task (Scenario 1)

	Time Step						
	1000	1500	2000	2500	3000	3500	4500
Task	1	25					
	2	25	25				
	3		25				
	4			44	11		
	5				22	16	10
	6					20	
	7						16
	8						15
	NONE		50	6	17	14	9

These result show that maximizing local entropy does lead to the creation of self-organized behaviors. Agent decisions create fairly balanced teams across known system targets. These self-organized teams create periods of stability easily seen in the order graph. In addition, points of critical system transition, both to higher and lower levels of

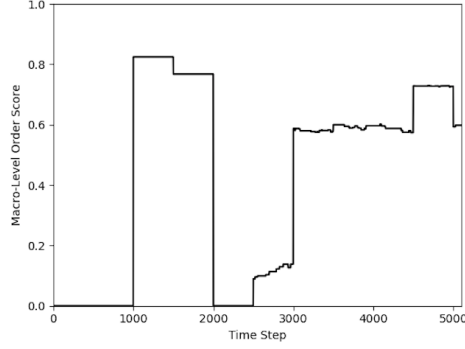


Figure 15. Macro-level order over time for UAV Scenario 1.

system order, are collected. Although possible transition points were known *a priori*, they were dynamic where the appearance or disappearance of targets was based upon random chance. The drop at time step 1,500, captures a change in local agent entropy calculations as one target disappeared as a new one appeared. Although the resulting distribution was the same, i.e. even distribution across two tasks, the metric captured the dynamic change which would have been missed by pure distribution metrics. Finally, this graph shows how the order metric picks up on other agent behaviors, such as refueling. Gradual decline or increase in system order due to agent refueling is captured during regions of relative stability.

Scenario 2.

In this scenario, agent sensor and communication ranges were limited to 500 and 250 pixels respectively, increasing the likelihood of uneven task distribution in the system (see Table 6). All other settings such as possible transition points and fuel randomization remained the same.

Table 6. Control and Independent Variables for UAV Scenario 2. All speeds are in pixels per time step, with ranges in pixels. Fuel and life metrics decremented by 1 at each time step.

Control		Independent	
Refueling Center	(500, 950)	Fuel Level	[3,000, 8,000]
Visual Radius	360 degrees	Sensor Range	500
Velocity	0.50	Comm Range	250
Swarm Size	50	Target Life	[500, INF]

Table 7. Number of Agents Per Task (Scenario 2)

Task	Time Step						
	1000	1500	2000	2280	2500	2750	3000
1	24	17					
2	24	16	14	12			
3		17	9	18	37	36	36
4			16	19			
5					8	11	2
6							3
NONE	2		11	1	2	3	9

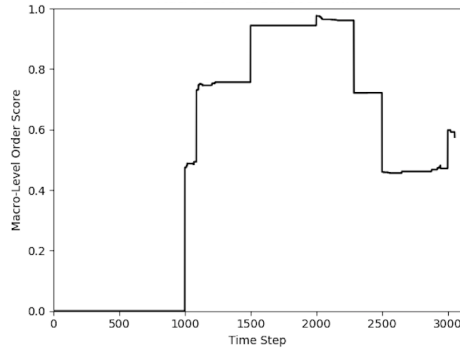


Figure 16. Macro-level order over time for UAV Scenario 2.

Table 7 and Fig. 16 show task distribution and macro-order scores over time for the second scenario. As before, when there is a fairly even distribution of labor across multiple tasks, the macro-level order increases towards 1. Additionally, the macro-level order drops when tasks disappear and agents reassign themselves. The system reaches a peak macro-level order score at 2,200, when the agents disperse evenly across three known tasks plus an equal number engaged in refueling or random search. At 2,280, a sudden drop occurs as the random agents discover and engage tasks in the environment. This time step shows how sensitive the metric is to changes in system structure. At 2,500, there is another drop due to the loss of a task, resulting in large imbalance. As stated earlier, this occurrence will

be fairly common when agents have restricted sensor ranges. Finally, macro-level order appears relatively stable until 3,000, when Task 6 appears, resulting in a change in agent distribution.

Scenario 2 again highlights that agents seeking maximum local entropy scores creates self-organized behaviors that place order upon the system. Furthermore, the uneven distribution of agents across tasks creates a more chaotic, or less orderly system, reflected by lower order scores associated with time steps where the population of agents was skewed across targets. However, the order metric is still able to identify periods of stability and points of criticality.

Scenario 3.

In the final UAV scenario, possible target appearance and dissolution occurred every 500 time steps at a 65 percent probability, creating a highly dynamic environment. All UAV sensor ranges were set to 500 pixels (see Table 8).

Table 8. Control and Independent Variables for UAV Scenario 3. All speeds are in pixels per time step, with ranges in pixels. Fuel and life metrics decremented by 1 at each time step.

Control		Independent	
Refueling Center	(500, 950)	Fuel Level	[3,000, 8,000]
Visual Radius	360 degrees	Sensor Range	500
Velocity	0.5	Comm Range	500
Swarm Size	50	Target Life	[500, INF]

Table 9. Number of Agents Per Task (Scenario 3)

		Time Step							
		500	1000	1500	2000	2500	3000	3500	4000
Task	1	17							
	2	17							
	3		17						
	4		17						
	5			26					
	6			21					
	7				46	13	21		
	8					17			
	9					15			
	10							18	24
	11							21	
	12								
	13								
	NONE	16	16	3	4	5	29	11	26

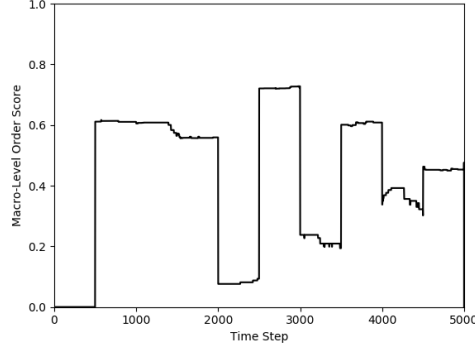


Figure 17. Macro-level order over time for UAV Scenario 3.

Table 9 and Fig. 17 show task distribution and macro-order scores for the third scenario. Clear periods of stability and points of criticality appear in Fig. 17. Starting at time step 500, the system enters into a period of relative stability as agents equally disburse across two tasks and random searching. A gradual decline at 1,500, occurs when agents engaged in random search begin finding tasks in the environment. A sharp drop in system order occurs at 2,000, as the system reduces from two tasks to one that attracts a large proportion of the agent population. After time step 2,000, increases and decreases in system order (i.e. points of criticality) align with known checkpoints for possible task creation or dispersement (every 500 time steps) with periods of stability appearing between these points. Small perturbations in system order that occur during these stable regions are due to small numbers of agents switching from random search or refueling to target engagement, or vice versa, with rises associated with agents disengaging from targets and dips associated with agents finding, and engaging, them. In this manner, both random search and refueling create another group structure in the simulation that impacts system order.

Game of Life.

The Game of Life is more stochastic than the UAV swarm with new cells being created and destroyed at almost every time step. Like the UAV simulation, periods of stability occur, usually as a result of the emergence of an oscillating pattern. Points of criticality,

however, are more difficult to detect unless the system experiences major changes in neighborhood structures. For example, a glider moving across will not change the system order. From the order metric's perspective, the system is in a constant or quiescent state.

Scenario 4 – R Pentomino.

R-Pentomino was selected for its chaotic start providing a challenge for the system order metric as points of criticality may be obfuscated by the noise of constant neighborhood creation and destruction in the early time steps.

The macro-order graph (Fig. 18) shows that although the R Pentomino pattern immediately enters a highly stochastic state, the order of the system, and number of stable neighborhoods, increases during the first 200 time steps. Then the system undergoes a long period of stability with minor perturbations occurring as the pattern plays out. Many of the early neighborhoods enter quiescent states with the noted changes occurring when new neighborhoods go through their lifecycle. At time step 1,130, the entire system enters a quiescent state.

Although able to track these rapid changes in structure and population size, the order-metric cannot pinpoint critical transition points with any amount of certainty. If one averaged every 10 time steps into an order value, only one period of transition, time steps 10 - 30, creates a negative growth state (i.e. the order decreases). The rest of the time, the system is increasing in order until about time step 200. Arguably, every time step in the early life cycle of the R Pentomino pattern is a critical point, only stabilizing once larger, stable, neighborhood structures appear.

Scenario 5 – Glider.

This work selected the glider pattern as it thwarted one of the entropy based metrics in [12]. In this scenario, the glider pattern moves across the screen until impacting the

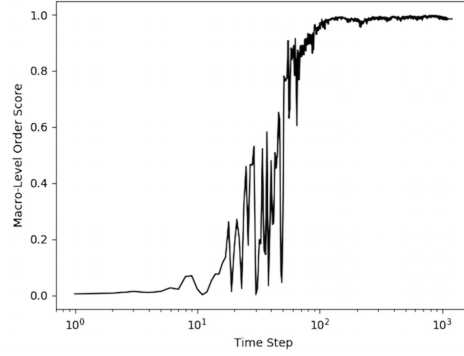


Figure 18. Macro-level order over time for R Pentomino.

boundary and creating a permanent block structure. The macro-level order metric (Fig. 19) clearly shows the boundary impact point at time step 230 and the resulting, permanent, block pattern from 231 onwards. As stated earlier, the order metric does not denote any difference in the glider's state until it comes into contact with the boundary and undergoes a transformation in size. The metric would have to be modified to be able to discern differences in system state due to glider movement where the structure composition fails to change (i.e. same number of connected alive cells) but the positions of the alive cells change. However, unlike the R Pentomino pattern, clear, critical transition points exist in the glider pattern, as indicated by the large drop in system order. The resultant shape, a four block square, is smaller than the original structure, and thus imposes less order on the entire system.

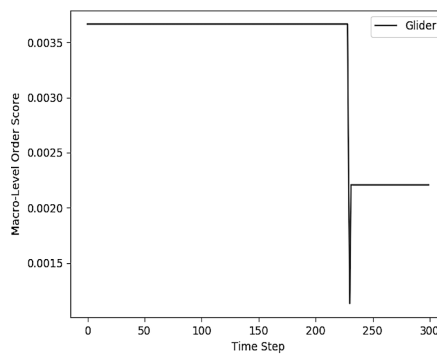


Figure 19. Macro-level order over time for a Glider.

Scenario 6 – Gosper Gun.

The Gosper gun is interesting as it has multiple subcomponents whose transformations continuously create gliders moving in a southeastern direction from the “gun” group. In other words, a stable pattern continuously creates new patterns *ad infinitum*. Graphing macro-level order values (Fig. 20) reveals some interesting patterns. First, the initial, stable patterns show clear oscillation features as they transform states. These transforming states then create a greater oscillation pattern as they produce gliders. The ramp up in glider production levels reaches its highest peak near time step 200. Afterwards, the entire system is an oscillating pattern of glider production and destruction as they impact the right most wall. The dips in the oscillation pattern represent the transformation and destruction of gliders against the right wall. A glider at time step, n , will collide with the wall and transform into a four block neighborhood. The glider at time step, $n + 2$, will then collide with the four block neighborhood, resulting in the ‘death’ of the cells and a drop in macro-level order. The next glider in formation will start the pattern all over again. Again, the order-level metric can detect changes in system order due to structural change and track periods of stability. However, critical points are still difficult to pinpoint in the early stages of system macro-order growth.

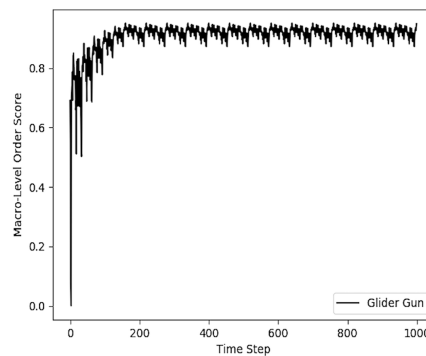


Figure 20. Macro-level order over time for a Gosper Gun.

4.6 Discussion

These results highlight some interesting points. First, agents maximizing local neighborhood entropy can impose order on the system through self-organizing behaviors, implying agents do not require global knowledge to create an adaptable system. This observation is in-line with what is known about decentralized natural systems. Ants and bees have limited memories and sensors [107, 109], basing decisions upon current environmental stimuli which includes input from other agents. Some ant species in particular have shown a threshold based response where an ant's action decision is impacted not by pheromones but by the number of times it encounters another type of ant [109]. In other words, evolution tuned ant behaviors to reward a division of labor but how separate species individually select which task to complete is more complex than previously thought. Bees engage in similar behaviors; however, bees use more complicated visual signaling methods, such as the waggle dance [107] to share information. However, in both cases, agents share knowledge locally. Granting agents in an engineered system full global knowledge, or the ability to estimate millions of future states, is both contrary to observation, and, as this work shows, unnecessary to produce intelligent, collective behavior.

Additionally, local neighborhood entropy and population distribution produce a macro-level order metric sensitive to micro-level dynamics incorporating agent interactions, e.g. communication in the UAV swarm, and structure, such as connected cells in the Game of Life. It also highlights temporal dynamics without having to calculate them explicitly. For example, the small perturbations of order during periods of relative stability in the UAV swarm correlate to agent refueling and random search behaviors tracked by task distribution versus individual fuel levels or a forced refueling function. The critical points seen in the UAV graphs also highlight the appearance and disappearance of tasks. Although coded to occur at certain time steps, the macro-level order metric is sensitive to changes in the environment resulting in observable deviations from current agent behaviors and neighbor-

hood structure. One could leverage this in a detection scheme where a sliding scale average is used. Any value that is two to three standard deviations from the current window mean would alert an observer to a possible change in system behavior. These changes could be desired self-organized behaviors, such as teams forming for target coverage, or possible malicious acts. Using the UAV example, if the expected behavior is the even distribution of UAVs across known tasks and yet multiple UAVs continue to fly in random patterns or converge to one specific target, this could indicate some type of control failure or hijacking.

The central thread for tying micro and macro-level behaviors together is assigning meaning to the entropy metrics used. Here, macro-level order indicates the formation, or dissolution, of agent structures and the population assigned to them. If the macro-level order rises, teams are forming, if the macro-level order decreases, teams are dissolving, making the metric useful for interpretation. Additionally, reducing system states to a generalized attribute [94] allows the use of entropic methods without having to estimate possible probability distributions or densities. Attribute-based entropy methods allow for specific probability measurements.

However, limitations for this approach do exist. One, not all systems can be easily reduced to a set of even priority tasks with homogeneous agents or neighborhood structures. For heterogeneous agents, one would need either separate entropy measures or some type of hybridized approach to estimate the global macro-order related to each type of agent. Otherwise, some underlying micro-pattern changes may be lost in the “noise” of a combined estimation. As Haghnevis [49] argues, simply combining three components with entropy values, $x + y + z = 1$, tells one very little as any of the components could hold those particular values. Furthermore, task prioritization is ignored in this method. It is possible to give tasks a higher priority but then a threshold parameter needs to be introduced to enforce division of labor at some point. Without this threshold, all available agents would flock to the highest priority task, ignoring all others, which is probably an undesirable feature. Fi-

nally, this approach does not guarantee team-to-task optimality. The agents select tasks based solely upon local entropy levels without regard to current fuel status or distances, leaving some tasks untended as the agent that selected that task returns to refuel and in some cases, the target dissipates before the agent arrives when another agent could have reached it sooner.

4.7 Conclusion

This chapter presented a macro-level order metric that incorporates both local entropy scores and macro-level agent population distributions. Experimental results showed that the macro-level order metric is able to detect both periods of stability and points of criticality in two dissimilar domains. The estimated system order reflected the state of self-organized behaviors in the system as well as enabling detection of stability and critical points during the system's lifecycle. Although an exploratory method, the metric showed sensitivity to micro-level agent interactions on a scale detectable to an outside observer. In this manner, this metric can be used for the detection of emergent patterns of self-organization and decomposition inside a complex system. This work showed that micro-level entropy scores correlate to increasing and decreasing macro-level order imposed upon the system that align with Holland's hierarchical view that macro-level patterns rely on changing micro-level patterns [55] in complex adaptive systems.

Additionally, this chapter showed how to construct a multi-task capable, multi-agent systems using informational entropy. Leveraging entropy-based work from [116, 105, 141, 90], this work used entropy to guide agent decisions in a swarm tasked with covering multiple dynamic targets. Results showed that agents could quickly switch their targets, and by extension their functional niche, based on local entropy measures. The physiological differences imposed by entropy measurements resulted in a robust and adaptable swarm.

Experiments in this chapter built homogeneous populations with physiological differ-

ences driven by local entropy calculations, resulting in agents forming sub-swarms that were able to find and follow targets as they randomly appeared in the search space. Using entropy, these agents were able to functionally align to different targets, creating a uniform distribution of agents to known targets. The behaviors exhibited by the UAV agents in the presented simulations provide evidence that informational entropy helps establish a robust and adaptable system.

Furthermore, results support the overall hypothesis that speciation aligns agents into sub-goal oriented *niches* that, once aggregated, accomplish system level goals. Here, agents established multiple niches, e.g. agents in search mode, tracking mode, as well as a niche for each target as it appeared. In this manner, one sees the hierarchy of niches formulating macro-level behaviors. With the metric developed here, one can visually see how stable the niches were as targets entered, and left, the simulation, as well as times when agents had to refuel. The refueling behavior required the system to be robust, as other agents had to realign and cover any targets that were now without a tracking agent. Finally, using the metric in Conway's Game of Life to track the creation, and destruction, of cell structures, highlights the metric's generalizability.

V. Entropy-Based Team Self-Organization with Signal Suppression

5.1 Introduction

Inline with research questions two and three, this chapter continues to explore speciation and entropy-based methodologies to create robust, multi-task capable systems.¹ The focus of this chapter is to compare the performance of *heterogeneous* teams, i.e. agents possess physical differences, to *homogeneous* teams, i.e. agents possess the exact same sets of sensors and abilities, in the same domain. As in previous chapters, entropy guides agent decision making; however, a new signal suppression technique is introduced to move agents toward new, macro-level goals. This chapter further explores the impact of agent speciation and entropy-influenced physiology on multi-agent systems.

5.2 Motivation

Computational systems composed of several independent subsystems often rely on central coordination to achieve shared goals. However, as the size of the system increases, centralized controllers become overwhelmed by increasing complexity, resulting in the need for decentralized control approaches. This becomes even more challenging when different tasks need to be tackled concurrently to achieve an overall goal. In order to meet this challenge, this chapter presents a novel self-organizing approach using an entropy-maximization task selection technique that incorporates signal suppression behaviors found in bee colonies. The technique enables autonomous agents to dynamically select and form teams to tackle different but complementary tasks. These teams are formed exploiting local knowledge gained from on-board sensors and information shared among nearby agents. At the same time, the suppression signal limits information sharing to a specific number of

¹The majority of this chapter was published in the proceedings of the conference on Artificial Life (ALIFE 2019) held in Newcastle, United Kingdom, July 2019.

agent required to accomplish the task.

The author tests the technique in a version of the online multi-object k -coverage problem [35]. In this domain, all agents can sense targets within their respective sensing range. If an agent has a target within its sensing range, it may elect to follow it, once followed, the target is considered *covered*. Together, the collective of agents aims to maximize the number of targets covered by k agents.

The chosen domain presents several challenges. One, the set of targets is not fixed and may change over time. This means targets can arise or disappear at any moment, creating the need for agent robustness and flexibility. Two, agents do not know the number or locations of targets within the global environment, constraining agent knowledge. Three, both targets and agents are not static, moving freely through the environment, challenging the agent collective to continuously find and cover targets.

This problem gives rise to an interesting agent dilemma: should an agent follow a target in order to maximize the number of agents covering it or continue to search through the space in order to increase the probability of detecting new targets. To overcome this dilemma, this work introduces multiple teams for different purposes. First, agents are divided into two teams where the first team's emphasis is on target discovery while the second team focuses on coverage of those targets. Second, the covering agents dynamically team up to ensure each target is provisioned with k agents. The relation between the team sizes and the amount of targets and its effect on the overall team performance is previously unexplored. Therefore, this work also studies the impact of three team and agent properties: heterogeneity, team size, and sensor ranges, on team performance in accomplishing target detection and coverage goals.

Specifically this chapter is interested in the following three research questions:

1. Can an entropy-maximization technique, aided by bee colony based signal suppression, be used to inform a collective autonomous decision making process, where

agents decide whether to cover a target or continue searching for other targets? Furthermore, what is the effect of such an approach on dynamic team formation and selection?

2. Is a homogeneous team of tracking agents, employed with target detection and coverage, more effective than heterogeneous teams composed of tracking and observer agents each tasked with different goals, i.e. detection and coverage of targets over time?
3. What is the impact of the communication and sensing ranges of individual robots on the achieved rates of detection and k - n coverage when employing homogeneous and heterogeneous teams?

5.3 Related Work

This work investigates the potential benefits of team work through the lens of heterogeneous and homogeneous teams engaged in a two-dimensional, k - n coverage domain. The methodology presented here tests three team properties: an entropy-based agent decision function, group size, and sensor ranges, and their impact on goal accomplishment.

The k -coverage problem combines the Cooperative Multi-robot Observation of Multiple Moving Targets (CMOMMT) problem introduced by Parker and Emmons [104] and the k -coverage problem associated with sensor networks proposed by Huang et al. [59]. The problem's main premise is for a population of agents to discover random targets in space and assign k agents to them, where k is a predetermined number of agents. Researchers have used various methods to try and solve the k -coverage problem. For example, Werger et al. [135] assigned weights to each target based on the number of robots and targets in the environment, and agents would broadcast their eligibility to engage the target to allow for team coordination. However, these assignments occurred with *a priori* knowledge which is not realistic in real-world environments.

To overcome this limitation, Jung et al. [67] proposed enabling robots to calculate target weights at run-time which allowed robots, who shared this information with the collective, to distribute themselves accordingly through the search space. Kolling et al. [78] allowed agents to request help from others to create longer loiter times for agents over assigned targets. They also added a signaling feature where agents could request help in covering targets as they appeared.

In a similar manner, bee colonies use direct signaling to marshal help from other bees. As foragers return to the colony, they recruit other bees to help collect resources via vigorous waggle dances [107]. In engineered systems, Tolba's agents used signaling to gather underwater search vessels towards a discovered sunken target, while agents in Beard et al. [8] used signaling to coordinate flight plans and rendezvous points. Similarly, in a search and rescue domain, robots in Jennings et al.'s [66] experiments used signaling to ask others for help in accomplishing their tasks. Signaling in both natural and engineered systems serves as a coordination medium over which information, such as target locations and agent capabilities, are shared. In this work, agent signaling is used to inform, i.e. share target locations, as well as to influence other agent behaviors, akin to bees suppressing the waggle dances of other bees ([80]).

Theraulaz et al. [127] suggested that individual agents should both be able to autonomously select teams, as well as switch teams when necessary. Lewis et al. [84] showed that team performance can be increased if individual agents change behaviors based on the current system state. Although Gross et al. [47] argued against agent specialization, and showed that teams could accomplish goals without it, many natural systems do evolve specialized agents.

Wilson [139] theorized that environmental pressures created the conditions for the emergence of specialized castes in ant colonies. It is logical to assume that such specialization serves the needs of the ant colony in a manner that is worth the energy required

to create different types of ants in the same colony. This work tests the effectiveness of heterogeneous teams, i.e. teams with specialized agents, and homogeneous teams, i.e. teams without specialized agents. Agent behaviors rely on both agent suppression signals as well as an entropy-based decision process.

Using entropy to guide agent decisions is a relatively recent approach in multi-agent systems. Agents in [105], [141], and [90], used entropy calculations to guide immediate agent decisions. Each showed that some measure of system entropy could enable dynamic agent coordination and team building. King and Peterson [76] used entropy maximization to guide target selection in an Unmanned Aerial Vehicle (UAV) search domain. Similar to this work, simulated UAVs searched a 2D space for moving targets. UAVs would swarm to detected targets within sensor range; however, the UAVs were not held to specific k - n coverage demands. Instead, agents evenly divided themselves among available targets by polling local neighbors and selecting the target that resulted in the highest increase in local entropy. As even task coverage is inadequate for the k - n coverage problem in this work, a suppression signal - a behavior inspired by bee colonies - was incorporated into the agent decision process.

The suppression behavior creates a dynamic decentralized-control mechanism to prevent over-provisioning. Unlike previous works, such as [51], [40], and [87], both agents and targets move through the space. As targets randomly appear, and disappear, *a priori* based approaches, e.g. [34], cannot be used, as agent teams must adapt at run-time to detected targets. The environmental dynamics requires teams to self-organize into sub-teams to be able to meet both goals of k - n coverage and target detection. Dynamic team self-organization can be accomplished by specialized agents, or encoded behavior responses to local phenomena.

The methodology presented here extends previous work by incorporating entropy-maximization and agent signaling suppression techniques to create a decentralized-control mechanism.

This mechanism allows dynamic team, and sub-team, formation in response to current environmental conditions. Additionally, it explores the potential benefits of specialized teams in a dynamic, k-coverage domain.

5.4 Problem Statement

At its core, this is a k -assignment problem coupled with resource constraints. Agents must position themselves in the environment to effectively detect a large number of targets whose appearance, time and position, are unknown *a priori*. Furthermore, once agents detect targets, they must meet agent-target distribution (i.e. k - n coverage) requirements while still positioning themselves to find, and cover, any additional targets that may appear.

Formally, given a set of targets, $O_t = \{o_1, o_2, \dots, o_n\}$, at time t and a set of agents embodied in mobile robots, $R = \{r_1, r_2, \dots, r_m\}$, agents must discover and provide a k - n coverage of targets who appear with a probability of $\rho = 0.60$ and duration of $\lambda = [500, INF)$. The point, $l_i(t) = (x_i, y_i)$, represents an agent's or target's location in 2D space at time t . All agents and targets move through the 2D space at a set velocity, $v_i(t)$, and heading, $\omega_i(t)$ with all agents being assigned a random initial heading at t_0 . Agents have a sensing range σ and may communicate within a set range χ .

Two types of target discovery and tracking agents exist: trackers and observers. Tracker agents can discover and track, i.e. cover, detected targets within sensor range, or those shared by observer agents. Observer agents can only discover targets. Once a target is discovered, Observer agents broadcast the location of the target to all agents within range. Tracker agents add the target to their target lists, and then decide whether to cover the target or not. Tracker agents never broadcast the location of any discovered targets to their neighbors.

Agents *detect* targets when they are within the sensing range of the robot, i.e. for a distance d_{ij} between a target i and an agent j and $d_{ij} < \sigma_j$. Further, $det_i = 1$ if $\exists j \in R(d_{ij} <$

σ_j) indicates that a target i has been detected by at least one robot. Agents are unaware of the total number of targets in the environment. They are only aware of targets within their sensor range, or those shared by observer agents. The first goal of the set of agents is to maximize the number of detected targets in the environment:

$$det_{max} = \max \sum_{i=1}^n det_i \quad (10)$$

The second goal of our agents is to maximize the number of targets being covered by k agents:

$$kcov_{max} = \max \sum_{i=1}^n kcov_i. \quad (11)$$

A target i is covered $cov_{ij} = 1$ by a robot j if the agent decides to follow the target and $kcov_i = 1$ if $\sum_{j=1}^m cov_{ij} = k$ or 0, otherwise.

5.5 Entropy-based Team Self-organization

Our proposed approach aims to enable agents to achieve two goals: detect targets as they appear and provide k - n coverage of detected targets. Agents create k - n coverage of targets through entropy-maximization and signaling. When a target is detected, agents poll all agents within their communication range to create an agent-centric distribution of agents to targets. Using Shannon's entropy equation (Equation 12) [118], agents select the target that creates the highest entropy value for their neighborhood.

$$H = - \sum_i p_i \log_2 p_i \quad (12)$$

Specifically, in Equation 12, p_i is calculated by dividing the number of known agents currently assigned to target i by the total number of agents within sensor range. Each agent adds itself to the number of agents assigned to both the target and population for each known target, resulting in a local entropy score for that target. The agent then selects the

target that produces the highest resultant entropy score, i.e. maximizes its local entropy score.

The shortfall with this method is agents will create a uniform distribution across known targets which, for k - n coverage problems, is insufficient. To overcome this, agents already engaged in a target with maximized coverage send suppression signals to other agents within their communication range. When an agent receives a suppression signal, it drops that target from its detected list, allowing it to either engage other known targets, or continue to search for undiscovered ones.

Tracker, observer and target agents possess different update rules that follow Reynold's flocking method ([110]) with respect to separation and goal seeking behaviors. Tracker agents update their position by moving towards any detected targets, $g_i(t)$, while avoiding other tracking and observer agents, $sep_i(t)$ (13). Observer and target agents update their positions similarly, only adding the separation vector, $sep_i(t)$, to their update rule (14). Resultant vectors were capped by a max movement value of 0.688 pixels per time step. The behavior of an agent a_i at time t_i can be described as:

$$l_i(t+1) = l_i(t) + v_i(t) + sep_i(t) + g_i(t) \quad (13)$$

$$l_i(t+1) = l_i(t) + v_i(t) + sep_i(t) \quad (14)$$

$$\omega_i(t+1) = \omega_i(t) + \alpha_i(t) \quad (15)$$

Each agent autonomously updates their internal heading vector, $\omega_i(t)$, with respect to its desired heading, $\alpha_i(t)$, capped by a max steering force of 0.05 at each time-step.

5.6 Experiment

Experiments ran a mixed number of trackers, [5-25], and observers, [0-15], through 1,000 simulations of 5,000 time steps a piece. Teams without observers, i.e. observers =

0, were considered homogeneous. Heterogeneous teams were composed of any number of tracker and observer agents where the number of observers was greater than zero. Both tracker and observer agents were allowed to move freely about a 1,000 by 1,000 pixel grid. Tracker sensor and communication ranges were initialized at 50 pixel intervals [100,200] with observer sensor ranges spanning 50 pixel intervals [200-350]. Up to two targets from a ten target pool were randomly selected and placed into the environment every 500 time steps. Targets freely floated through the grid guided solely by one avoidance rule: keep a 25 pixel distance between it and any other object. All agents and targets were bounded by the grid. Initial positions for all agents and targets were randomly assigned. See Table 10 for control and independent variables.

Table 10. Control and independent variables.

Control:	
Visual Field	360 degrees
Velocity _{max}	0.688 pixels/time-step
Steering Force _{max}	0.50
Targets	10
Prob Target Appearing	0.60
Prob Target Disappearing	0.50
Independent:	
Tracker Team Size	5, 10, 15, 20, 25
Observer Team Size	0, 5, 10, 15
Tracker Sensor Range	100, 150, 200
Observer Sensor Range	200, 250, 300, 350

5.7 Results

Figures 21 and 22 show the average detection and coverage rates achieved by n tracker and m observer teams across all sensor ranges. Results indicate that the entropy maximization with signal suppression method creates dynamic teams capable of high target detection and coverage rates. Specifically, the method leads to dynamic, self-organization of sub-teams capable of providing the desired k - n coverage rates (i.e. $k = 3$ in this case).

Furthermore, it becomes clear that the suppression signal used in our approach, prevents over-provisioning targets with more than k agents. However, distinct differences in team behaviors appear as the number of observers increases across all sensor ranges.

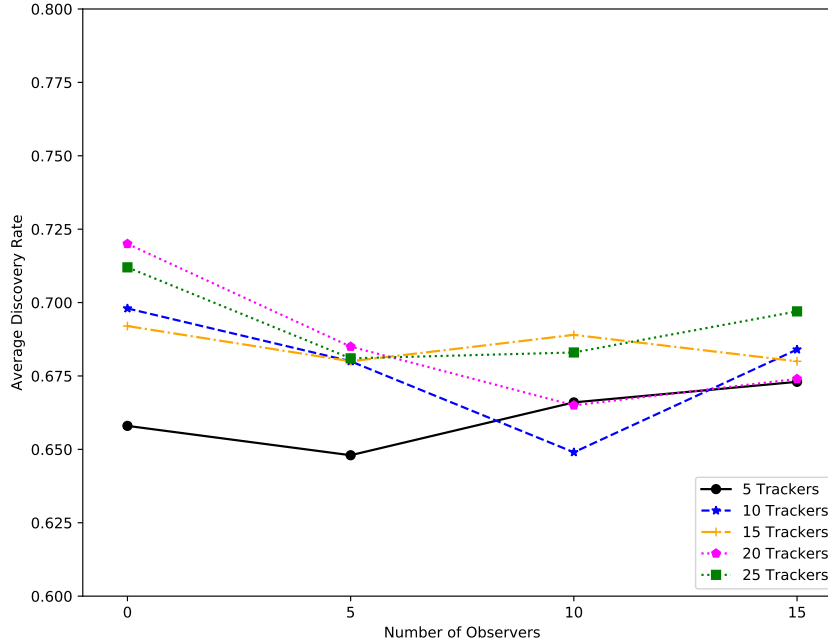


Figure 21. Average detection rates across all sensor ranges with teams of n trackers and m observers (x-axis), $k = 3$.

Results show a positive correlation between heterogeneous teams, i.e. those composed of specialized observer and tracker agents, and k - n coverage rates. However, as the number of observer agents increase in these teams, the detection rate immediately decreases. As more observer agents are added, a slow, positive increase in detection occurs, but never reaches the success of homogeneous teams in the space.

Although this result may be surprising, a closer inspection of heterogeneous team behavior explains the disparity. As observer agents locate targets and propagate target locations to tracker agents within sensor range, tracker agents begin clumping together. Information sharing ends up creating dense pockets of tracker agents pursuing targets in the

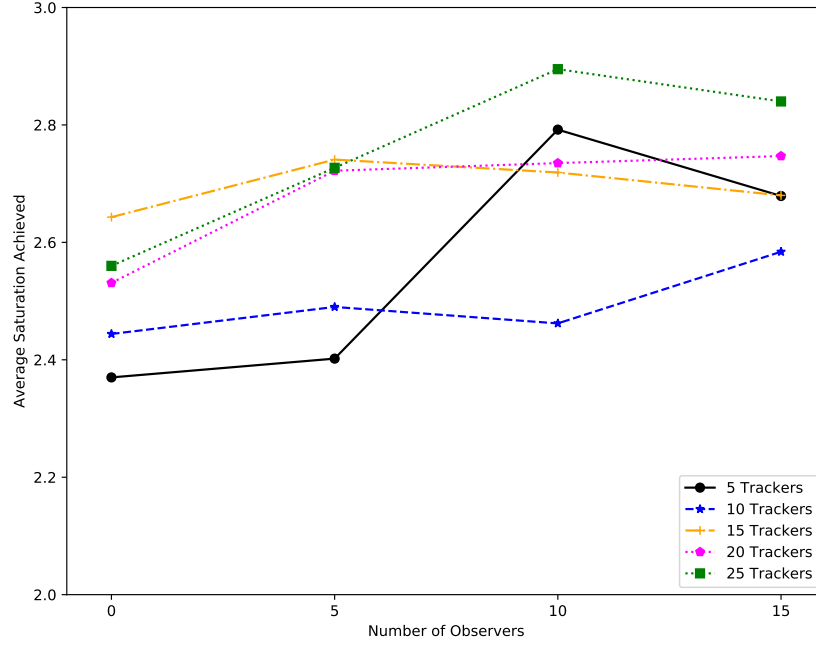


Figure 22. Average saturation rates across all sensor ranges with teams of n trackers and m observers (x-axis), $k = 3$.

same vicinity. The tracker agent sensor and communication range limitations cause a delay in suppression behaviors. The delay ends up altering the course of multiple tracker agents, resulting in multi-agent sensor overlap, decreasing the sensor range of the entire population. However, these pockets do result in higher coverage rates as more agents reside in an area to cover discovered targets.

Homogeneous teams, not sharing new target information, spread out across the grid, increasing the rate of task detection. Furthermore, signal suppression occurs immediately as sensor and communication ranges are equal for all agents, which avoids agent clumping. Homogeneous teams, however, do have the lowest coverage (i.e. saturation) rates, which makes sense given their population density is lower across the entire domain. The authors conclude that these behaviors create a trade-off between homogeneous and heterogeneous teams in the k - n domain.

From Figure 21, it is clear that heterogeneous teams do not approach the detection rates of homogeneous teams even when outnumbering homogeneous teams by a margin of 5:1. Coverage rates are a little more difficult to generalize as some heterogeneous teams peak with teams that contain 10 observers, while others, require less (e.g. teams with 15 tracking agents), or more (e.g. teams with 10 tracking agents). However, all experience a positive increase in coverage rates as the population size of observers increases. Notably, target coverage rates do approach the desired $k = 3$ rate. Theoretically, a pure 3:1 rate would require 30 tracker agents to cover 10 tasks; however, the results indicate that the desired 3:1 rate can be accomplished with fewer tracker agents. This shows that both the introduction of observer agents and entropy-based, dynamic team formation can reduce the number of tracker agents required to cover certain regions.

Table 11 presents the top three performing teams for detection and coverage tasks across all sensor ranges. Overall, homogeneous teams make up 47% of the highest performing detection teams with a 74.47% average detection rate compared to heterogeneous teams that achieved a 72.16% detection rate. This result again highlights the difficulty heterogeneous teams have in balancing detection rates with required coverage costs. For coverage rates, heterogeneous teams comprise 94% of the top performing teams with an average 3.00 saturation rate, with teams of 25 tracking agents, with 10 to 15 observer agents, making up the majority of the top performers. These results further solidify the argument that heterogeneous teams, i.e. teams with specialized agents, can create a benefit to the entire population with respect to task completion.

In the next set of results, Boneabau's Fixed Threshold Model (FTM) [13] (Equation 16) created a correlation coefficient for detection and saturation rates.

$$T_{\theta}(S) = \frac{s_i^n}{s_i^n + \theta_i^n} \quad (16)$$

Normally, agents use the FTM to decide which task to engage. It is used here for its

Table 11. Top 3 teams and their detection and saturation rates across all sensor ranges. TR and OR represents sensing range of tracker and observer agents, respectively. High 3 D and High 3 S the three highest results for detection and saturation (k-n coverage), respectively. T and O stands for the number of used tracker and observer agents, respectively.

TR	OR	High 3 D	T	O	High 3 S	T	O
100	200	0.65	10	0	2.986	25	10
		0.64	10	5	2.963	15	10
		0.63	15	5	2.937	20	5
100	250	0.9	25	0	3.11	25	15
		0.75	25	15	2.801	25	10
		0.72	20	0	2.775	5	10
100	300	0.68	25	0	3.479	25	10
		0.67	20	10	3.066	5	10
		0.67	10	0	2.981	15	10
100	350	0.78	20	10	3.014	15	10
		0.78	20	0	2.93	25	10
		0.75	10	15	2.92	25	5
150	200	0.68	20	0	2.847	15	15
		0.67	25	10	3	25	10
		0.67	20	0	2.835	25	15
150	250	0.69	25	5	3.392	20	10
		0.69	25	0	3.073	20	15
		0.68	25	15	3.002	15	5
150	300	0.7	15	0	2.867	25	15
		0.69	25	0	2.815	25	0
		0.69	20	10	2.769	15	5
150	350	0.91	20	0	2.81	25	15
		0.8	10	5	2.793	20	15
		0.77	15	10	2.793	20	5
200	200	0.85	10	0	3.152	25	10
		0.75	25	15	3.131	15	10
		0.73	15	15	3.081	20	5
200	250	0.67	20	15	3.196	15	15
		0.66	25	15	3.007	15	5
		0.66	25	0	2.956	20	10
200	300	0.77	25	0	2.842	25	15
		0.76	15	0	2.792	20	15
		0.75	20	15	3.789	20	10
200	350	0.88	20	0	3	25	5
		0.85	15	10	2.976	25	10
		0.78	20	5	2.828	10	0

mathematical properties to yield a correlation value along the interval, (0,1]. The stimulus parameter, s_i , represents the team detection rate with parameter, θ_i , representing the normalized team k -n coverage value. If detection dominates, then the FTM produces values close to 1, while if k -n coverage dominates, then the FTM produces values close to 0. If s_i and θ_i are close to equal, then the FTM produces values near 0.5. Once graphed, these

values yield an intuitive understanding of how the team performed with respect to both detection and coverage tasks.

Figures 23-27 present correlation plots using Bonabeau’s FTM function for all possible teams types across all sensor range combinations. These plots provide a measure of how well certain team combinations work towards both goals simultaneously, i.e. target detection and coverage. For example, in Figure 23, a heterogeneous team of 5 tracking agents and 5 observer agents with sensor ranges set to 150 and 200 pixels respectively provides an almost perfect balance (correlation = 0.500) of detection and coverage. These plots also provide insight on teams that excel at either target detection or coverage. Again from Figure 23, a team of 5 tracking agents and 20 observer agents with sensor ranges of 100 and 300 provides the best coverage rate, while a homogeneous team of 5 tracking agents with a sensor range of 150 provides the best detection rate.

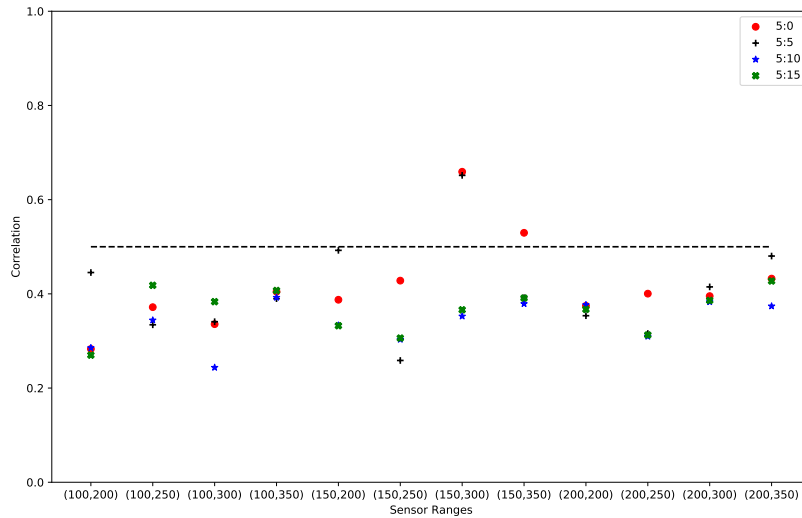


Figure 23. Correlation between detection and k-n coverage for mixed teams of 5 trackers and n observer agents across tested sensor ranges, with $k = 3$.

A closer inspection of these correlation plots yields some interesting behaviors with respect to the difference in sensor ranges in heterogeneous teams. As far as detection rates, there is a general increase in detection rates as the difference between tracker and

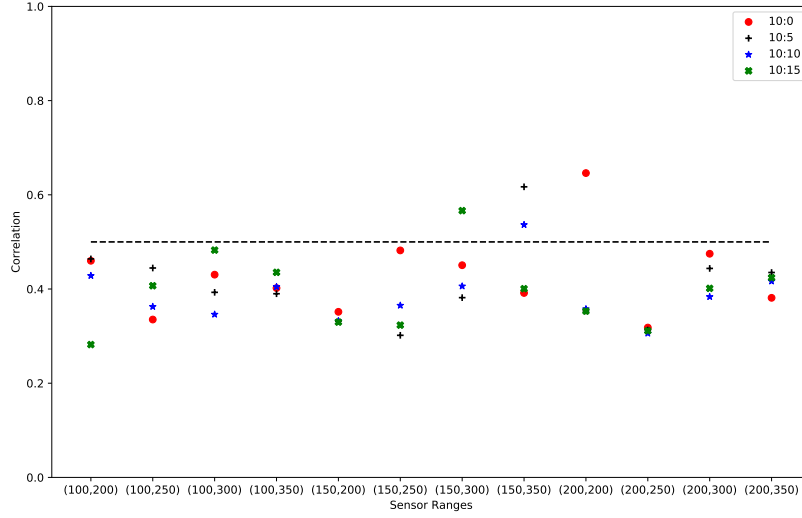


Figure 24. Correlation between detection and k-n coverage for mixed teams of 10 trackers and n observer agents across tested sensor ranges, with $k = 3$.

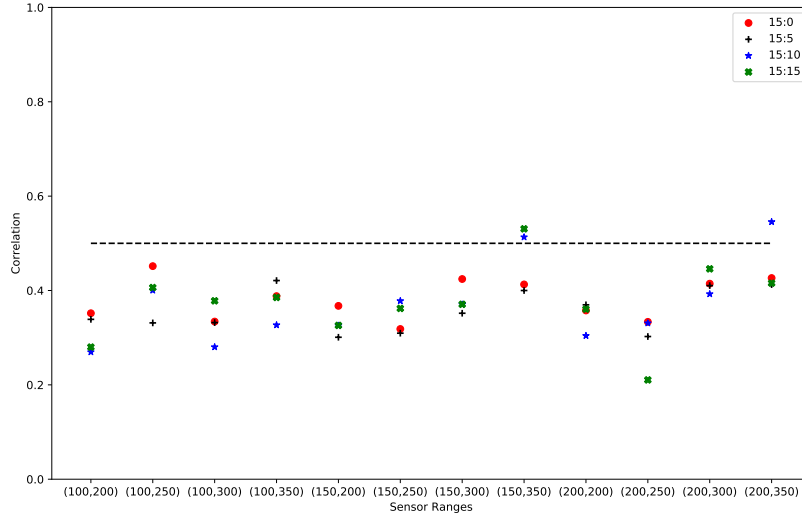


Figure 25. Correlation between detection and k-n coverage for mixed teams of 15 trackers and n observer agents across tested sensor ranges, with $k = 3$.

observer sensor ranges grows. Although this increased detection rate never exceeded the detection rates of the homogeneous teams, one could hypothesize that the detection rates of heterogeneous teams would eventually match, and possibly exceed, the homogeneous

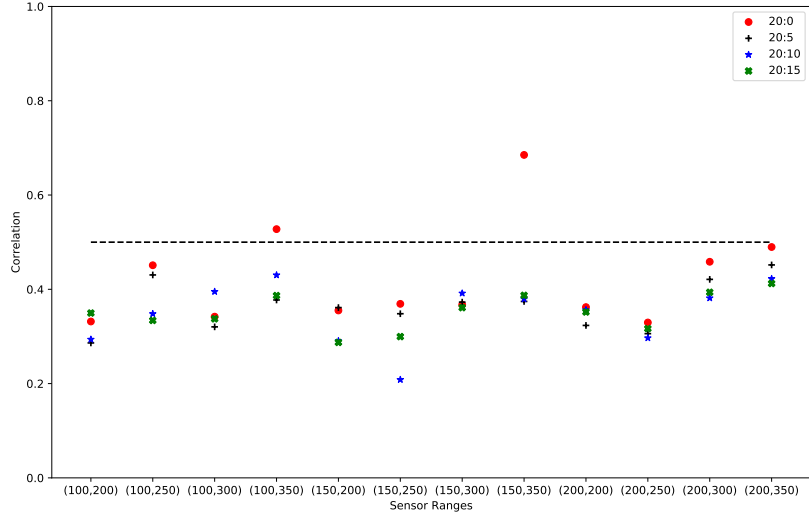


Figure 26. Correlation between detection and k-n coverage for mixed teams of 20 trackers and n observer agents across tested sensor ranges, with $k = 3$.

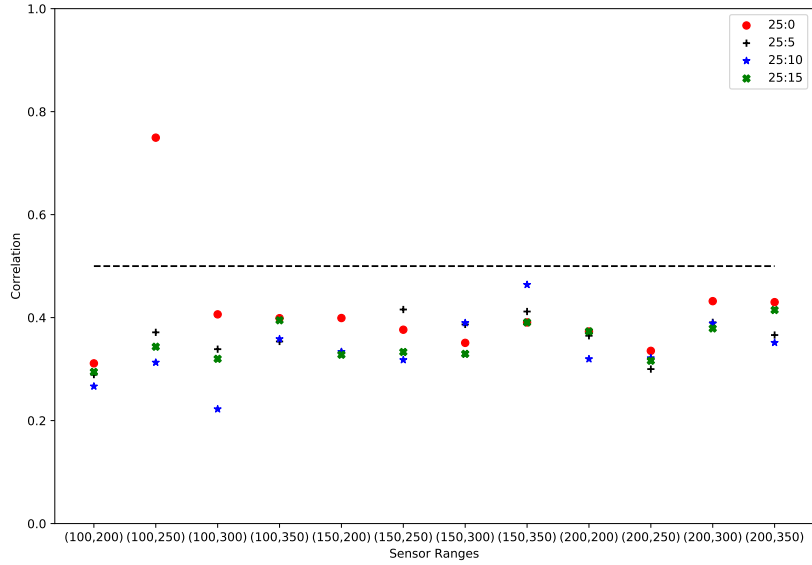


Figure 27. Correlation between detection and k-n coverage for mixed teams of 25 trackers and n observer agents across tested sensor ranges, with $k = 3$.

teams. Coverage rate results were more difficult to decipher.

In the 5 and 10 tracker team scenarios (Figure 23 and 24), correlation measures do not

show any sensor range patterns other than a chaotic, oscillatory nature. However, starting at 15 tracker teams (Figure 25), a couple of heterogeneous teams begin showing more stable behavior. In the 15 tracker team scenario, both the homogeneous 15 tracker team and the heterogeneous 15 tracker, 5 observer agent team exhibit small oscillatory behaviors but the highs and lows remain numerically close, i.e. fairly stable. Similarly, in the the 20 and 25 tracker team scenarios (Figures 26 and 27) both 20:5, 25:5, 20:15, and 25:15 heterogeneous teams exhibit the same, mostly stable, behavior across all sensor ranges. It is possible that larger teams are less affected by the differences in sensor ranges between tracker and observation agents, yielding more stable dynamics and behaviors, lending further validity to the conclusion that specialized teams are better than non-specialized ones in multi-task domains.

5.8 Discussion

The experimental results showed that homogeneous and heterogeneous teams excel in different tasks. Homogeneous teams were able to discover more targets on average than heterogeneous teams. In fact, almost all homogeneous teams experienced a decline in detection rates when observer agents were added to the team. Heterogeneous teams dominated the coverage task. The majority of top performing teams with the coverage task goal ($k = 3$) were heterogeneous with a mix of 25 tracking agents and 10 to 15 observer agents. These results indicate a trade-off between target discovery and coverage for heterogeneous teams.

As discussed earlier, heterogeneous teams create dense pockets of tracking agents, reducing the effective sensor range of the team in 2D space. However, these pockets lead to better saturation rates over time as more tracking agents are able to move towards a newly discovered target. Additionally, as the agents are in closer proximity to one another, suppression signals keep coverage teams to the desired k - n ratios, in turn minimizing the wast

of resources per target. As no clear team compositions dominated any others, no conclusions can be drawn about which teams would fare better in certain sensor range combinations, although one can conclude that heterogeneous robot teams outperform homogeneous teams in target coverage.

Results clearly show that agents using a local entropy maximization technique, coupled with a signal suppression strategy, produces robotic teams capable of high target detection and k - n coverage rates. Entropy-maximization based techniques lead to even agent distributions, showing that the signal suppression strategy is pivotal for the technique to reach the desired k - n coverage ratios. Other techniques, such as agent auctioning, would have to be tested against the entropy-maximization technique before one can draw any clear conclusions on effectiveness. Additionally, these future tests would need to track and compare the number of messages passed between agents to establish whether the technique is more efficient.

The impact of communication and sensing ranges of individual robots on the achieved detection and saturation rates appears mixed. All of the heterogeneous teams exhibited oscillatory behaviors with no apparent correlation to sensor range differences between tracker and observation agents. Although, at larger team populations, some heterogeneous teams showed less variability in their correlation values, meaning the amplitude of their oscillations were smaller. The authors can only conclude that sensor ranges increases are positively correlated to the accomplishment of both target detection and coverage. However, no correlations were found in the differences between tracker and observer agent sensor ranges.

5.9 Conclusion

The methodology presented here showed that agent specialization provided benefits in the accomplishment of desired k - n coverage tasks. Homogeneous teams were able to out-

perform heterogeneous teams in detecting more targets. However, as the number of agents in the heterogeneous team grew, they closed the detection rate gap. These results indicate that a possible ratio of tracker to observer agents could exist in which heterogeneous teams are able to exceed homogeneous team accomplishment scores in both tasks.

Furthermore, experiments showed the validity of entropy-based agent decision functions. Entropy-maximization coupled with suppression signals created the conditions for the emergence of tracker agent sub-teams capable of both detecting targets and providing k - n coverage. Interestingly, teams using the entropy technique were able to meet the desired k - n coverage percentage (i.e. $k = 3$) without requiring the theoretical population to achieve the 3:1, agent to task ratio, showing a possible resource efficiency.

The experiments in this chapter showed how one can engineer, multi-level, hierarchical systems using both morphological and physiological speciation. Interestingly, results showed that the type of speciation impacts the overall performance of the system which provides an avenue of future research as there could be a possible link between system flexibility and agent plasticity. In other words, morphological agents could be less plastic, resulting in a more rigid system while physiology could change, *online*, resulting in more flexible, i.e. robust, systems. However, more experiments are needed to justify these observations.

VI. The Emergence of Division of Labor in Multi-Agent Systems

6.1 Introduction

This chapter extends methods found in earlier chapters to investigate the emergence of division of labor in complex systems.¹ Division of labor in natural systems enables resiliency in times of dynamic change. Researchers have shown that division of labor can emerge in homogeneous populations predicated on the system's environment and the distribution of agent task bias. This chapter demonstrates that the emergence of division of labor in homogenous populations is also impacted by agent decision functions, agent population size, and environmental constraints. Although homogeneous, populations evolve internal agent bias which instantiates physiologically different species inside the total population.

Results show, one, agent decision functions and population size have a significant impact on the division of labor scores, whereas, the influence of environmental constraints depends upon the chosen agent decision function. Two, results indicate that high division of labor scores do not necessarily translate to higher resource production, which, again, appears tied to agent decision functions. Three, although agent population size possesses a positive correlation to division of labor scores, agent decision functions play a more critical role in its emergence.

This chapter supports all three research questions as speciated populations must complete nine different tasks in a dynamic environment. The functional alignment of agents towards specific tasks creates niches that form desired, macro-level behaviors, in this case, the making of a resource. In experiments on robustness, populations undergo severe environmental change during runtime. Results show that speciated populations are both robust to environmental change and capable of completing multiple tasks concurrently.

¹The majority of this chapter was published in the proceedings of the SASO 2019 Conference held in Umea, Sweden, June 2019.

6.2 Motivation

The division of labor, where individuals perform different tasks simultaneously versus sequentially, is an important component in many natural systems [111]. Division of labor creates efficient use of agent labor and has been documented in natural systems such as ants [43], bees [64], and humans [16]. The emergence of castes, where groups of agents specialize in particular tasks, is a hallmark of the division of labor in these systems [139, 33]. Castes usually evolve to create labor efficiencies, such as larger ants in the species *Solenopsis invicta* that carry larger prey back to the colony, excavate more dirt, and even break down seeds to make food more accessible to other ants [139]. However, most ant species only have one large worker class [139], indicating that division of labor can also emerge in homogeneous populations.

In engineered systems, division of labor can be defined as the efficient management of agent labor to complete tasks that may be costly or time consuming [82]. The cost of a task, with regards to resource requirements or time, creates different categories of tasks. System agents must learn to coordinate their actions to efficiently complete these tasks. Works delving into system efficiency propose an advisor, or a centralized controller [121] to achieve such results. The focus of this work, however, is not on efficiency or problem optimization, rather, it explores the conditions that effect the emergence of division of labor in multi-agent systems. If one accepts the view that division of labor emerged in natural systems to facilitate efficient management of labor, then it stands to reason that emergent division of labor is a desired property for decentralized, multi-agent systems. If one can uncover the underlying properties of division of labor, one can engineer systems in which it is an emergent property.

In spite of a long research history, a universal theory on how the division of labor emerges in both social and non-social societies remains unexplained. However, division of labor appears directly related to agent interactions among themselves and their environ-

ment [43], guided by social behaviors [64] and agent temperament [53]. Environmental interactions can forge unique behavioral traits, such as ant rescue techniques in some sand-dwelling ants [58], as well as agent task selection through resource availability [60] and spatial locality [130, 64]. Each of these properties serves as inputs to an agent's *decision function*, impacting the emergence of division of labor.

While many researchers focus on optimizing the behavior of multi-agent systems by utilizing methods such as auctioning [119, 89] or predetermined task specialization [95], few analyze the agent decision function [18], instead defaulting to Bonabeau's fixed-threshold model (FTM) [13, 32]. Few stop to ask if the function is appropriate for the task or environment at hand? This article seeks to understand what properties, including the choice of an agent decision function, an engineered system must have that leads to the emergence of division of labor in engineered systems composed of homogeneous agents.

Division of labor provides engineered systems with many benefits. One, it occurs without a centralized control authority, meaning an engineered system with this property could self-organize responses to dynamic task and environmental changes, creating a resilient system [103]. Two, the emergence of the division of labor leads to greater system efficiency. Agents self-assigning tasks, without pre-programmed heuristics and minimal inter-agent communication, leads to more efficient use of system resources, creating higher productivity [33]. The key question to answer is what properties must a system employ to create an organic division of labor, and benefit from its emergence?

Previous work [33] concentrated on the emergence of division of labor by biasing agents and their evolution. Although bias is an important part of a system, biological researchers have found other factors such as *agent behaviors*, e.g. social constructs and temperament [53, 64], *population size* [43, 98, 83], and *environmental constraints*, e.g. resource availability [60], agent-task proximity [130] and task cost [33], all influence division of labor emergence. This work sets out to answer the following questions with respect to the divi-

sion of labor in engineered systems:

1. How do agent behaviors, expressed as an agent decision function, impact a system's division of labor?
2. Does an increase in population size yield an increase in the division of labor and, therefore, increase a system's productivity?
3. How do environmental constraints, i.e. task availability and cost, influence a population's division of labor?

By answering these questions, one learns how to engineer systems where the division of labor is an emergent property, leading to labor efficiencies without forcing agent specialization *a priori*.

The work presented here investigates multi-agent system properties that lead to the emergence of the division of labor. It empirically studies the effects of agent decision functions, population size, and environmental constraints on the evolution of division of labor in a homogeneous population working in Adam Smith's Pin Factory [120].

Results show that both agent decision functions and population size have strong correlations to the division of labor scores. In most cases, population size has a positive correlation to the division of labor; however, some experiments indicate that a maximum-population density to task ratio may exist. Agent decision functions appear to impact both division of labor and resource production, indicating that the choice of the correct agent decision function for the environment is critical to system success.

6.3 Background

Biologists have extensively investigated how the division of labor emerges across multiple species, from ants [43] to humans [98]. Division of Labor (DOL) emerges as multiple agents work towards common goals. It is theorized that DOL is an evolutionary choice [64] and can lead to the emergence of castes in insect societies [130, 139, 138]. Although no

all-encompassing theory for DOL emergence exists; several common properties found in research appear to affect its appearance in natural systems.

Gordon [43] proposed three questions on the emergence of the division of labor in biological collectives. First, how much do agent behaviors and attributes, e.g. agent bias, physical adaptations, etc., contribute to the dynamics of the system? Second, how does the size of the system determine its dynamics? And, third, how do the internal system dynamics react to and bring changes to the environment? Gordon argued that agent interactions and the environment drove dynamic task allocation in biological collectives and felt that answering these questions would lead to a better understanding of how DOL emerges in those systems.

Duarte [33] investigated agent bias by developing an evolutionary method that studied the impact of internal agent thresholds on the development of long-term specialization, i.e. the emergence of castes, in homogenous populations. Duarte showed that a high distribution of internal-agent threshold values (i.e. agent bias), as well as high-cost task switching, resulted in evolutionary branching and the emergence of DOL. Additionally, Duarte showed that across colonies there was a positive correlation between specialization and the amount of work achieved. Importantly, that a colony starting from unspecialized workers could self-organize into differentiated workers. These results fall in line with Beshers and Fewell [11] and Jandt and Dornhous [63], who independently showed that task specialization appears to emerge due to differences in agent behavior (agent bias). In this work, the influence of agent bias is coupled to the agent decision function, showing how the function, not just its parameters, creates different division of labor scores. Furthermore, Section 6.5 refutes the noted positive correlation between agent specialization and the amount of work achieved in [33].

Multiple biologists have looked at Gordon's second question about group size. Ulrich, et al. [132] showed that division of labor can emerge as group size increases in homoge-

neous populations. Nakahashi, et al. [98], contended that DOL is likely to evolve as a population's size is large and learning is important for group survival. Jeanson, et al. [65], tested the theory that a positive relationship exists between group size and the emergence of DOL based on the changes in the group dynamics. As group size increased, they generated higher levels of DOL. However, this correlation does not appear to be a universal truth. Oliveira and Campos [29] showed that size-complexity does not always hold between the division of labor and population size, indicating a possible diminishing returns affect between population size and DOL. Jeanson, et al.'s [64] work showed similar results as agent populations with higher internal bias experienced declines in DOL as population-task-ratios increased over a 5:1 ratio.

Other empirical researchers, such as Holbrook, et al. [53], Nakahashi and Feldman [98], and Hollis [58], explored the impact of the environment and coevolution among species with respect to DOL and task availability. Each showed that the environment impacted DOL through the presence of resources. For example, Holbrook, et al. [53] found that docile and passive members of the spider species, *Anelosimus studiosus*, which engage in particular tasks aligned with temperament, would complete less-desirable tasks if no other spiders existed to complete them. Interestingly, Holbrook, et al. also found that DOL emerged in colonies with as few as five spiders, indicating that even small homogeneous groups can benefit from DOL. Hollis [58] showed that the environment can influence an agent's task knowledge. Specifically, that certain sand dwelling ants knew how to rescue others from ant lion traps, an unobserved behavior in non-sand dwelling species. Nakahashi and Feldman [98] showed how DOL developed in hominids based on the demand for a skill (environment driven) and the time devoted to specializing that skill. These works show that internal dynamics influence, and are influenced, by the surrounding environment, and lead to evolutionary DOL traits.

The background literature indicates that the division of labor is a multifaceted system

property. In order to build a system that contains division of labor, one must determine how each of the properties mentioned above impact a group’s division of labor.

6.4 Methodology

This work leverages Duarte’s [33] research methodology and pushes it further by showing how an agent’s decision function, population size, and environmental constraints, affect DOL. Researchers such as Robinson [111], Leighton [83], Ulrich, et al. [132], and Duarte [33] all noted that behavioral actions, population size, task switching costs, and the environment individually create conditions for agent specialization. It is reasonable to hypothesize that if agent-bias values influence DOL, so would the agent’s decision function. Furthermore, based on biological research findings, we contend that population size should show a positive correlation with DOL scores. In addition to these properties, our work looks at the impact of environmental constraints, both temporal and resource based, on DOL. Finally, this work takes an exploratory look at how DOL is related to system resiliency by evaluating the performance of two colonies, equipped with different decision functions, as they experience a dynamic change in environment.

Evolutionary Method.

We devised a set of experiments using Duarte’s evolutionary colony method [33] to test the impact of agent decision functions, population size, and task availability on DOL scores. Each experiment began with 50, pre-initialized colonies populated with N workers, where N equaled the population size currently under study. Each worker received a stimulus value, θ_i , for each task, t_i , randomly drawn from a uniform distribution along the interval (0,1]. The stimulus value, θ_i , represents the agent’s internal bias towards selecting that particular task. At each time-step, t , agents used one of two assigned decision functions (i.e. models): Bonabeau’s fixed threshold model (FTM) or the entropy based linear

threshold model (LTM). Bonabeau's FTM [13] (Equation: 17) is widely used in biological modeling.

$$T_{\theta}(S) = \frac{s_i^n}{s_i^n + \theta_i^n} \quad (17)$$

In the FTM, the probability of performing a task is based on the internal agent task threshold, θ , the related task stimulus, s , and a steepness threshold, n . Accordingly, if $s \ll \theta$, the probability of performing the task is almost 0, while if $s \gg \theta$, then it is close to one [13]. In each experiment, task stimuli, s_i , changed at every time step according to $s(t+1) = s(t) + \delta_i - \alpha_i * A(t)$, where a task's increase in stimulus, δ_i , when not selected is offset by a decrease in stimuli, α_i , upon selection. $A(t)$ equals the proportion of agents currently working task t . Internal agent stimuli, θ_i , is randomly selected from the interval $(0, 1]$ and remained constant through the colony's lifecycle. The parameters α_i , δ_i and n were set to 3, 1, and 2 respectively.

The LTM is based on previous work with entropy-based agent decision processes [105, 141, 90, 76] and is the second decision function (Equation 18). The LTM uses task entropy, H_i , task stimulus, s_i , and the agent's internal task stimulus, θ_i , to produce a sum for a task. The agent selects the task with the highest sum.

$$f(s) = H_i + s_i + \theta_i \quad (18)$$

Task entropy, H_i , was calculated using Shannon's entropy (Equation 19) derived in [118]. The update rule for stimuli was changed to $s(t+1) = s(t) + 0.5$ if not selected, and $s(t+1) = s(t) - 0.5$ if selected. Stimuli values were capped between $[0,1]$ as they could quickly overwhelm agent bias, θ_i .

$$H_i = - \sum_i p_i \log_2 p_i \quad (19)$$

A task's entropy, H_i , was calculated by adding the current agent to the number of agents currently assigned to task i and dividing by all agents in the simulation, yielding p_i . Tasks with lower numbers of assigned agents have entropy values close to 1, while heavily saturated tasks have values close to 0. Task entropy provided a tie-breaking mechanism for any agents who had similar task stimuli and bias values for multiple tasks. Although a task could have a low entropy score, agents could still select it if task stimulus and/or task bias values were high enough.

Domain.

Adam Smith's Pin Factory served as the testing domain. In Smith's Pin Factory, workers must complete up to eighteen different tasks to create one pin [120]. Smith noted that if twenty workers accomplished these tasks in a lockstep manner, then they would collectively produce about four-hundred pins in a day. However, if these same workers specialized, where each worker concentrated on a smaller set of tasks, then production grew to an extraordinary 48,000 pins a day [120]. Smith theorized that DOL led to the increase in resource production.

Although Smith's Pin Factory can be considered a theoretical domain by today's standards, it shares attributes with contemporary multi-agent problem domains. The primary problem facing the factory is efficiently managing worker (i.e. agent) labor with respect to the time and resources necessary to complete different jobs (i.e. tasks). Domains such as the General Pick-up and Delivery problem [115] and the Tea Processing Warehouse [136] both have pools of workers that must fulfill specific tasks. In the Tea Processing warehouse, agents must deliver full and empty bins to different tea-processing machines to

grind, parch, and dry tea. The goals for the system must include flexibility, high throughput and reliability [136]. Importantly these tasks have an associated cost, e.g. time to deliver bins, and constraints, such as load capacity. Lacking a central advisor, agents must self-organize their actions, i.e. divide their labor, in response to dynamic task requests as they arrive.

Smith’s Pin Factory is a similar, but less complex domain. A pool of homogeneous workers must complete a series of tasks to produce a resource, i.e. a pin. The domain possesses multiple attributes to test the impact of agent decision functions, population size, and environmental constraints on DOL. First, it is a multi-task environment where each task can have both time and resource constraints. Second, ‘pin’ production provides a measurable system outcome with an easily formulated fitness function. Third, one can assign colonies different agent decision functions, allowing for direct comparisons of their performance and impact on DOL as population size and environmental constraints change. Finally, ‘pin’ production provides a measure in which to build correlations between DOL scores, resource production, and system resiliency.

6.5 Experimental Analysis

Pin Factory.

In our simulated Pin Factory, a homogeneous population of software agents, $A = \{a_1, a_2, \dots, a_n\}$, is tasked with producing as many pins as possible for a set time duration. An agent, a_n , can be defined by the tuple, $(f(x), t_n)$, where $f(x)$ represents its decision function and t_n represents its current task. At each time step, an agent must select a task from a set of nine, distinct tasks, $T = \{t_1, t_2, \dots, t_9\}$. As agents are homogeneous, factory agents possess the same internal decision function, either FTM (17) or LTM (18), and can accomplish every task. The set of tasks, T , require resources to complete. When a task, t_n , is completed, resources for task, t_{n+1} , immediately become available. Only when task, t_9 , is completed,

does the population produce a ‘pin.’ Unless otherwise noted, a set of ten resources for task, t_1 , was added at every time step to ensure workers had at least one task available.

Environment.

Environment variables, *rich* and *constrained*, refer to task availability. All tasks were available to agents at every time-step in *rich* environments, while in *constrained* environments, tasks required resources. As noted in [33], task switching costs (i.e. time to complete tasks) can influence the emergence of division of labor. In time-to-task length one (TT1), environments, every task took one time-step (ts) to complete. In time-to-task length variable (TTV) environments, tasks had different time-step requirements and were assigned by the following rule: $t_1 = 1, t_2 = 1, t_3 = 1, t_4 = 2, t_5 = 3, t_6 = 3, t_7 = 4, t_8 = 2, t_9 = 1$, where $t_1 = 1$ is interpreted as task one takes one time-step to complete, $t_4 = 2$, requires two time-steps to complete, and so on.

Simulations.

Each simulation followed the same process. After loading the initial 50 colonies, a colony processed as many pins as possible in a 360 time-step epoch. Agents selected a task to accomplish in a random order at each time-step. Once complete, the simulation ran the next colony through the same process. Each colony received a fitness score upon completing its epoch. The fitness function divided the number of pins produced by 48,000 - the number of pins Smith stated could be produced by 10 workers - creating a percentage, ρ . The simulation then added ρ percentage of workers from the colony to a population pool.

Workers selected for the pool underwent both crossover and mutation procedures outlined in [33]. Note: workers were pulled randomly from the population, meaning individuals were not individually selected for fitness, rather, it provided an indirect fitness function

on colony production; avoiding individual agent optimization. Finally, the simulation created 50 new colonies (i.e. the next generation) by randomly selecting N workers from the pool, with N equaling the number of workers the colony should contain.

Data Analysis.

Gorelick, et al.'s division of labor statistic [44] measured the degree of agent or group specialization with respect to tasks. According to Gorelick, et al., one calculates the division of labor by placing agent task selection frequencies in a data matrix of agents (rows) and tasks (columns). The matrix is normalized so all columns sum to one. Shannon's entropy is then used to calculate the entropy of individuals across tasks and mutual entropy (see Equation 20) for the entire matrix. Dividing mutual entropy by Shannon's index (Equation 21) provides a division of labor score, $[0,1]$, with 0 meaning division of labor is absent. Division of labor scores were averaged over all colonies per time-step.

$$I(X,Y) = \sum_{x \in X, y \in Y} p(x,y) \log \frac{p(x,y)}{p(x)p(y)} \quad (20)$$

$$D_{y|x} = \frac{I(X,Y)}{H(X)} \quad (21)$$

For all experiments, both the mean, μ , and 95% confidence intervals were calculated for outcomes. One-sided Student's T-Tests, with $\alpha = 0.05$, were used to verify statistical differences between distributions such that $[H_0 : A_0 = B_0, H_a : A_0 \neq B_0]$ where A_0 and B_0 represent two outcome distributions.

Division of Labor Experiments.

The first set of experiments tested how agent decision functions, population size, and environmental constraints affected DOL scores. The experiments iterated over each com-

bination of independent variables (Table 12), for a total of twenty-four experiments. For example, the first experiment set the population size to 9, a 1:1 ratio of agents to tasks, assigned the FTM function to every agent, and placed the colony in a resource restricted environment where tasks required resources but only one time-step to complete (environment = restricted, task length, TL = 1).

Table 12. Control and Independent Variables for Division of Labor Experiments.

Control		Independent	
Colonies	50	Population Size	[9, 27, 45]
Tasks	9	Decision Function	[Bonabeau, Linear]
Generations	1,000	Environment	[Rich, Constrained]
		Task Length	[1, Variable]

The next experiment changed the agent decision function to LTM, enabling a comparison of the agent decision functions for this particular scenario. Later experiments tested population size changes, e.g. 27 agents (3:1 agent-to-task ratio) and 45 (5:1 agent-to-task ratio), across agent decision functions, environments and task lengths, producing DOL scores for comparison. Finally, pin counts were averaged across all 50 colonies at each time-step to track resource production.

Data from the first set of experiments (Table 13) show that DOL emerges in every tested scenario. The LTM colonies score between 0.111 and 0.309, placing them on par with sweat bees, *Lasioglossum hemichalceum* (0.100), the social spider, *Anelosimus studiosus* (0.111), and the eusocial harvester ant, *Pogonomyrmex californicus* (0.25-0.37) [53]. FTM colonies score between 0.328 and 0.520, placing these artificial agents in the same category as harvester ant foundresses (0.32-0.71) and rock ants, *Temnothorax albipennis* (0.50) [53]. The impact on DOL scores due to the agent decision function choice is clear, with FTM DOLs scores exceeding LTM values in every scenario by almost a factor of two.

Population size shows a positive correlation with DOL scores as DOL scores increased as populations increased in the LTM colonies. FTM colonies show comparable results; however, a decline in DOL scores occur for FTM colonies in two 5:1 ratio scenarios (environment = restricted/task-length = variable, and environment = rich/task-length = one).

Population size results match theoretical biological research that shows a decline in DOL scores as ratios of agents to tasks increases should be expected at some point. Jean-son, et al.'s [65] work showed that when agent to task ratios went over 5:1, and they held task stimulus to values greater than or equal to one, ($s_i \geq 1$), that DOL scores decreased. Similarly, Oliveira and Campos' [29] experiments showed that increased task demand reduced colony specialization, i.e. reduction in DOL, providing further evidence that such behaviors could occur. Tannenbaum [124] noted in their experiments with pre-specialized agents that a maximum-population to maximum-efficiency ratio appears to exist as a result of agent density. In other words, at a certain point, only so many agents can specialize.

Table 13. The Effects of Agent Decision Functions, Population Size, and Task Constraints on Division of Labor Scores.

Ratio	Environment	TL	DOL (FTM)	DOL (LTM)
1:1	Restricted	One	0.328 +/- 0.034	0.131 +/- 0.031
1:1	Restricted	Var	0.354 +/- 0.027	0.143 +/- 0.030
1:1	Rich	One	0.330 +/- 0.361	0.112 +/- 0.034
1:1	Rich	Var	0.391 +/- 0.239	0.111 +/- 0.033
3:1	Restricted	One	0.356 +/- 0.083	0.191 +/- 0.051
3:1	Restricted	Var	0.473 +/- 0.032	0.220 +/- 0.043
3:1	Rich	One	0.465 +/- 0.318	0.243 +/- 0.041
3:1	Rich	Var	0.482 +/- 0.030	0.254 +/- 0.046
5:1	Restricted	One	0.418 +/- 0.052	0.256 +/- 0.060
5:1	Restricted	Var	0.461 +/- 0.036	0.287 +/- 0.051
5:1	Rich	One	0.445 +/- 0.055	0.288 +/- 0.048
5:1	Rich	Var	0.520 +/- 0.049	0.309 +/- 0.057

Task stimulus values (s_i) in our experiments could quickly reach, and exceed, one for both agent decision models, creating the conditions for a decline in colony DOL scores (i.e less agent specialization). It is more likely to occur in FTM-based colonies as stimuli values are not capped at one, which could explain why the behavior appeared in the 5:1 ratio. However, our work is the first to discover such a possible population-density to task-efficiency ratio for the fixed-threshold model. LTM colonies, whose task stimuli thresholds

Table 14. The Effects of Agent Decision Functions, Population Size, and Task Constraints on Average Pin Production.

Ratio	Env	TL	Avg Pins (FTM)	Avg Pins (LTM)	P.Value
1:1	Rest	One	812.991 +/- 162.510	1,663.359 +/- 299.193	< 0.001
1:1	Rest	Var	694.090 +/- 140.302	1,487.637 +/- 213.637	< 0.001
1:1	Rich	One	1,198.960 +/- 302.639	2,046.440 +/- 256.852	< 0.001
1:1	Rich	Var	871.937 +/- 200.031	2,104.6444 +/- 249.058	< 0.001
3:1	Rest	One	3,873.151 +/- 1,021.553	4,886.359 +/- 824.791	< 0.001
3:1	Rest	Var	2,997.934 +/- 574.279	4,741.898 +/- 657.871	< 0.001
3:1	Rich	One	3,807.006 +/- 581.009	3,876.883 +/- 515.546	0.009
3:1	Rich	Var	2,818.578 +/- 481.483	3,701.813 +/- 509.001	< 0.001
5:1	Rest	One	5,833.872 +/- 1,403.927	6,940.745 +/- 1,080.215	< 0.001
5:1	Rest	Var	5,489.508 +/- 967.388	7,226.541 +/- 1,011.141	< 0.001
5:1	Rich	One	6,728.885 +/- 1,777.138	6,343.426 +/- 896.636	< 0.001
5:1	Rich	Var	6,544.227 +/- 1,363.833	6,441.085 +/- 897.561	0.092

were tightly bound between zero and one, did not display a similar agent-to-task density threshold.

Task constrained environments, both temporal and resource driven, yielded some interesting results. For the FTM colonies, restricted-TT1 and restricted-TTV environments produced lower DOL scores than their rich-TT1 and rich-TTV environment counterparts. However, restricted-TTV environments always held higher DOL scores than the rich-TT1 colonies in the same ratio band, leading to the conclusion that task length (i.e. task cost), creates a demand for specialization, increasing the DOL score for the colony. The LTM colonies followed a different trend.

With the exception of the 1:1 ratio scenarios, LTM DOL values increased as environments and task lengths changed, with a resource-rich and variable task length scenario producing the highest DOL scores in the 3:1 and 5:1 scenarios. Here, access to resources provided the impetus for specialization in the LTM colonies akin to the FTM. The LTM 1:1 ratio scenarios appear to be an exception. Tannenbaum [124] argued that specialization is favored at lower population densities when resources are constrained and it is possible that small LTM-populations in a restricted environment become more specialized to meet system demand. However, this behavior does not result in higher resource yields (Table

14).

Figures 28 - 30 graph the pin production of the models over time. For clarity, every fifty time-steps were aggregated and averaged to produce these graphs. Table 14 holds the mean averages of all pins made over all time-steps for each scenario. Student's T-tests were used to compare distributions, only p-values greater than 0.001 are reported.

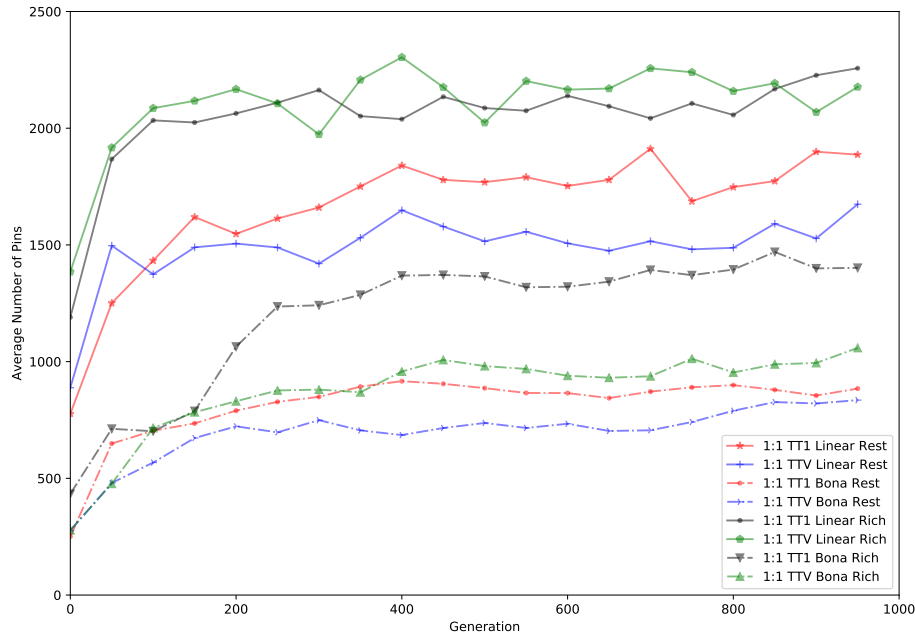


Figure 28. Average number of pins manufactured for 1:1 scenarios using the fixed threshold model (FTM) and linear threshold model (LTM).

LTM-based colonies outproduce FTM colonies in every 1:1 scenario (Figure 28) although they held lower DOL scores. Interestingly, LTM colonies in restricted environments outperform LTM colonies in rich environments in the 1:1 scenario. This indicates, for LTM colonies, that agent specialization is pressured by restrictive environments where agents must adapt to the environmental constraints (i.e. task length and resource availability). However, this only occurs in the 1:1 ratio, possibly indicating the influence of population size on resource production outcomes. Here, one also sees how DOL scores do not predict resource production. Although FTM colonies have higher DOL scores, they produce less pins on average than LTM colonies until the 5:1 scenarios.

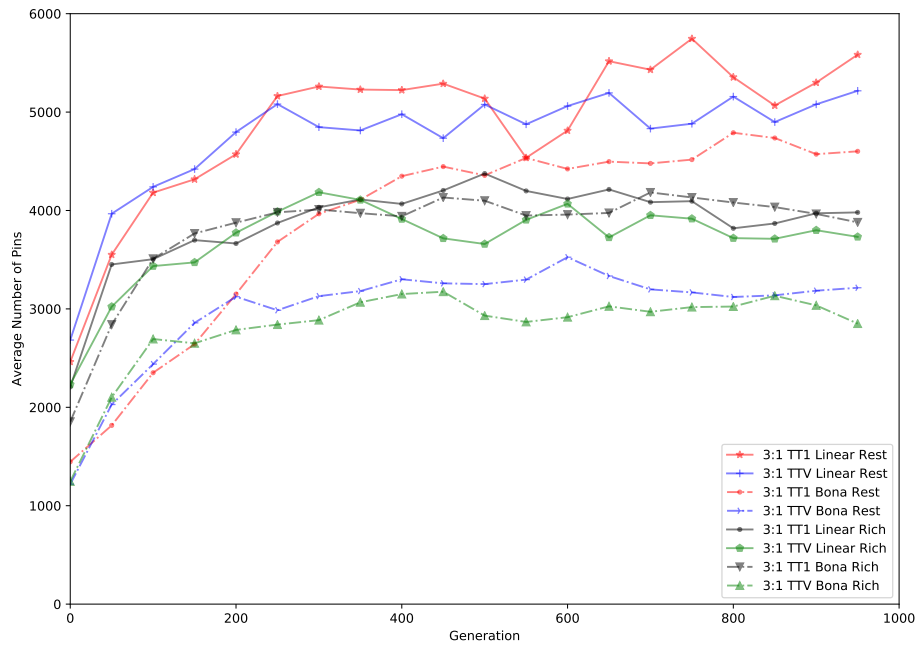


Figure 29. Average number of pins manufactured for 3:1 scenarios using the fixed threshold model (FTM) and linear threshold model (LTM).

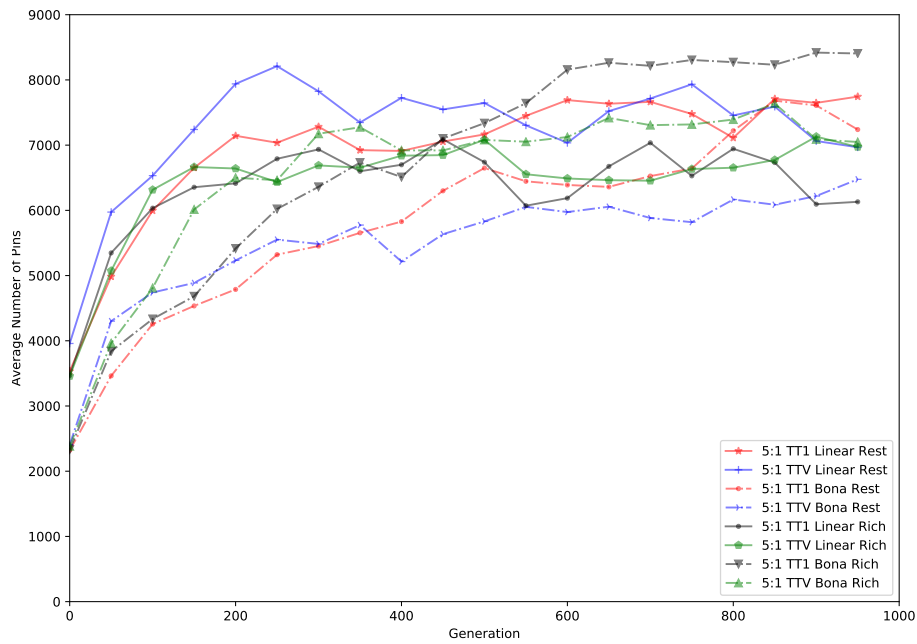


Figure 30. Average number of pins manufactured for 5:1 scenarios using the fixed threshold model (FTM) and linear threshold model (LTM).

The trend for LTM-colonies reverses once population size increases as they produce more pins in restricted environments at the 3:1 and 5:1 agent-to-task ratios than rich environments. FTM colonies follow a less conclusive pattern.

In the 1:1 and 5:1 ratio scenarios, FTM agents produced more pins in resource rich environments, than restricted ones. In the 3:1 ratio scenarios, the roles reverse with FTM colonies producing more pins in restricted environments than rich ones, in spite of higher DOL scores in resource rich environments, regardless of task length settings. These results make it difficult to predict how environmental constraints will impact the system as each agent based decision function appears to follow different patterns across every population size.

Generally speaking, both DOL scores and pin production increase as population size increases for the LTM colonies. As the agent to task ratio increases (Figure 29), both FTM and LTM colonies produce more pins and DOL scores also increase, indicating a positive correlation between population size and DOL. However, although correlated, this finding may not indicate causation. Comparing the DOL scores and resource output for both models, it is clear that LTM colonies in restricted environments outproduce the FTM colonies with lower DOL scores! FTM colonies do begin gaining ground on the FTM models in the 3:1 scenarios as both FTM colonies in rich environments keep pace with the LTM colonies in the same environment.

In the 5:1 ratio scenarios (Figure 30), FTM colonies begin outproducing LTM ones, especially in rich environments. As noted earlier, it appears that FTM colonies thrive in resource rich, low task cost environments while LTM colonies perform better in restricted ones. DOL increased for each type of colony, but, again, the DOL scores do not predict resource output. For example, FTM colonies with high task costs in rich environments have the highest DOL score of all scenarios tested; however, it is the FTM colonies with low task costs ($TL = 1$), and lower DOL scores, that produce the highest number of resources over

time. These results appear to support the conclusion that population size has a larger impact on resource production than DOL scores. However, it is the appearance of DOL that creates labor efficiencies and, thus, higher resource outputs.

Finally, inner-ratio comparisons yield some interesting observations about task constraints. Although noted earlier that resource restricted environments with higher task costs held higher DOL scores than the same colonies in resource-rich environments with low task costs (i.e. task cost = 1), they produced fewer pins on average. Consistently, for FTM colonies, resource restricted environments with higher task costs produced less than their low task cost counterparts. As before, the LTM showed different behaviors. Except for the 1:1 ratio, restricted environments produced higher resource yields than similar LTM colonies in resource-rich environments. This result leads to the conclusion that the LTM is better suited for resource-constrained environments, regardless of task cost, than the FTM when resource production is the goal while FTM works best in environments with large populations, plentiful resources, and low-task costs.

Resiliency Experiments.

Resiliency experiments tested the impact of agent decision functions and population size on system resiliency. Resiliency is the ability of agents to adjust with environmental changes [33]. The environmental shift used here caused a system bifurcation, which is a change in system state causing an abrupt shift in system behavior [38]. A resilient system will either recover to the original system behavior, or establish a new equilibrium, while a non-resilient system will collapse.

Each set of colonies began in a resource-rich, low-task cost environment. After 1,000 generations, all resources were removed and the environment changed to a resource restricted, high-task cost environment. Colonies ran for another 1,000 generations after the bifurcation. Both average pin output and division of labor scores were tracked for analysis.

Figures 31 and 32 show average pin outputs for FTM and LTM colonies respectively. A clear shift in system behaviors occurs at 1,000 generations for all FTM-based colonies. FTM colonies express a clear drop in pin production at the bifurcation point; however, in each agent to task ratio, the colony almost immediately reaches a new equilibrium, even developing positive trends in pin production for the 3:1 and 5:1 ratios. These results show that FTM systems are resilient to environmental shifts.

The LTM-based colonies (Figure 32) expressed different behaviors. The 3:1 and 5:1 LTM colonies remarkably end up outproducing their rich resource counterparts with the 1:1 ratio being the only exception. The 5:1 LTM colonies do experience a slow decline, i.e. a drop in production after generation 1,100 reaching a low point at 1,250. Afterward, the 5:1 colonies increase pin production, eventually restoring the system to the same production thresholds before the bifurcation. The 1:1 FTM results follow the trend noted earlier where a small population of agents became more specialized when resources became constrained, resulting in an increase in DOL but a decline in pin production (Table 15).

The shapes of the post-bifurcation behaviors when compared to pre-bifurcation behaviors are interesting. Pre-bifurcation shapes match those found in the first set of experiments where both agent decision models showed an almost logarithmic growth behavior. The post-bifurcation behaviors appear mostly positive linear in the case of FTM-based colonies, with the LTM colonies showing linear behaviors in the 1:1 ratio scenario, and logarithmic growth in the 3:1 and 5:1 scenarios. The differences in pre and post behaviors has to be contributed to the evolutionary state of the systems before the bifurcation, meaning colonies held evolved internal bias parameters at generation 1,000. These internal bias parameters probably impacted the resiliency of both systems more than the decision functions and population sizes alone. However, more experiments focused on internal bias would have to be accomplished to support this conclusion.

Although DOL scores in Table 15 show statistical differences pre and post-bifurcation,

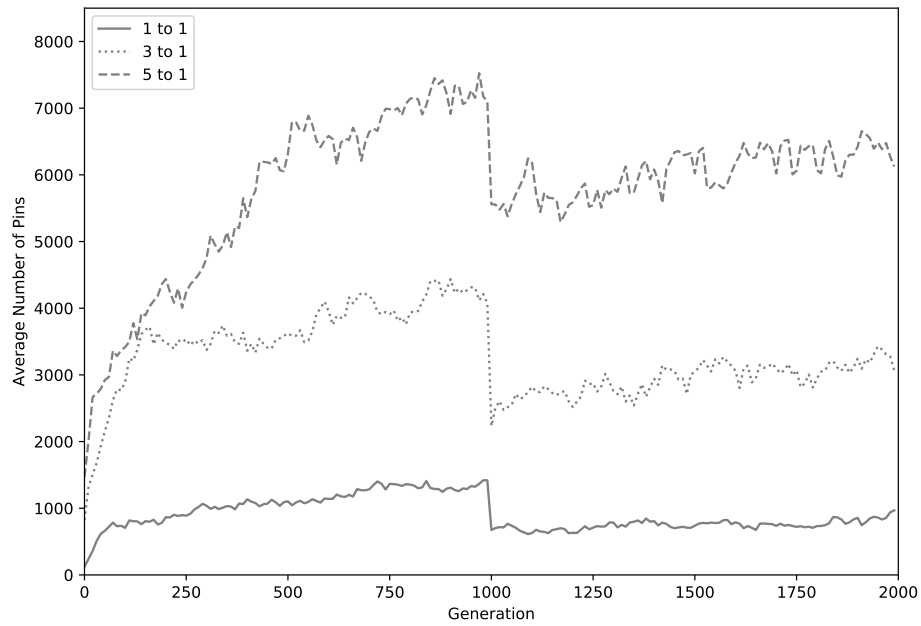


Figure 31. Average number of pins manufactured using the fixed threshold model (FTM) at 1:1, 3:1, and 5:1 agent to task ratios with a change in environment at generation 1,000.

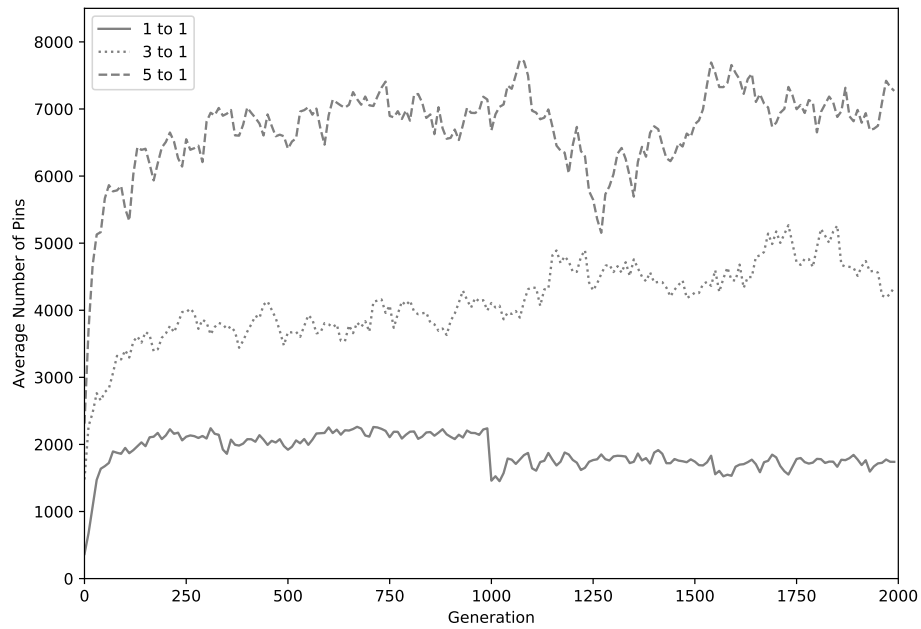


Figure 32. Average number of pins manufactured using the entropy-based linear threshold model (LTM) at 1:1, 3:1, and 5:1 agent to task ratios with a change in environment at generation 1,000.

the surprising results for the LTM-led colonies warranted closer inspection. The authors decided to narrow the focus on the system's DOL scores 200 generations before and after

the bifurcation (Table 16) since, by these points, the systems established equilibriums.

Table 15. Division of Labor scores for FTM and LTM colonies before and after bifurcation at generation 1,000.

Ratio	Model	DOL (0-1,000)	DOL (1,000-2,000)	P_Value
1:1	FTM	0.318 +/- 0.048	0.356 +/- 0.025	< 0.001
3:1	FTM	0.448 +/- 0.036	0.405 +/- 0.025	< 0.001
5:1	FTM	0.440 +/- 0.036	0.393 +/- 0.033	< 0.001
1:1	LTM	0.110 +/- 0.037	0.126 +/- 0.014	< 0.001
3:1	LTM	0.245 +/- 0.041	0.280 +/- 0.041	< 0.001
5:1	LTM	0.284 +/- 0.041	0.271 +/- 0.024	< 0.001

Table 16. Division of Labor scores for FTM and LTM colonies between 800 - 1,200 generations.

Ratio	Model	DOL (800-1,000)	DOL (1,000-1,200)	P_Value
1:1	FTM	0.326 +/- 0.014	0.349 +/- 0.014	< 0.001
3:1	FTM	0.416 +/- 0.026	0.387 +/- 0.0017	< 0.001
5:1	FTM	0.429 +/- 0.025	0.437 +/- 0.024	0.001
1:1	LTM	0.104 +/- 0.018	0.131 +/- 0.017	< 0.001
3:1	LTM	0.229 +/- 0.016	0.251 +/- 0.016	< 0.001
5:1	LTM	0.265 +/- 0.018	0.266 +/- 0.018	1.169

Table 17. Average pins produced by FTM and LTM colonies between 800 - 1,200 generations.

Ratio	Model	Avg Pins (800-1,000)	Avg Pins (1,000-1,200)
1:1	FTM	1,320.997 +/- 92.215	676.751 +/- 77.759
3:1	FTM	4,217.068 +/- 238.747	2,655.136 +/- 239.709
5:1	FTM	7,202.453 +/- 293.219	5,654.617 +/- 359.281
1:1	LTM	2,160.920 +/- 89.081	1,715.931 +/- 177.016
3:1	LTM	3,974.979 +/- 249.212	4,281.328 +/- 385.057
5:1	LTM	6,891.265 +/- 335.884	6,969.571 +/- 514.468

Tables 16 and 17 present the average DOL scores and pins from generation 800 to 1,200 for both agent model systems. By narrowing down the scope, a clearer picture of system states right before and after the bifurcation appears. Before the change, the 3:1 and 5:1 FTM models were actually outproducing their LTM counterparts. The superiority of FTM-based systems, once populations exceed a 3:1 ratio in rich environments, was noted in the

first set of experiments; leading to the conclusion the FTM agent decision function creates both high division of labor and resource yields in resource-rich, low-task cost environments with large populations.

LTM-based colonies show smaller declines in production at the 1:1 ratio, and actually increase pin production post-bifurcation in the 3:1 and 5:1 ratios. The DOL scores for LTM colonies increase for the 1:1 and 3:1 ratios but do not statistically change in the 5:1 ratio. In-line with previous observations, the LTM agents work best in resource-restricted environments with high-task cost. They also perform better than FTM agents in small population size to task ratios.

The apparent lack of correlation between DOL scores and the resource output of a system leads to the conclusion that DOL, and system performance, rely on both internal system properties and external influences. Agent decision functions and population size create the conditions for DOL to emerge. Task availability and cost, i.e. environmental constraints, control the DOL scale inherent to the system.

6.6 Discussion

Our results show that an agent's decision function impacts both the population's DOL score and its resource output. Although some colonies achieved high DOL scores on par with some natural systems, DOL scores did not appear directly correlated to resource production. Therefore, we cannot conclude that higher DOL scores are indicative of higher resource production rates or efficiency. Resource production rates appear related more to agent decision functions, population size and environmental constraints than DOL scores.

Results show a positive correlation between population size, DOL and resource production; especially in the LTM-based colonies. A maximum-population size to efficiency ratio appears in the FTM with a noted decline in DOL values between 3:1 and 5:1 agent to task ratios in two separate environments. LTM colonies never experienced such a drop

off; however, such a decline may appear with further increases in agent to task ratios. This finding indicates that DOL scores alone cannot predict resource output or labor efficiencies. However, large populations do increase resource production rates as agents work simultaneously on multiple tasks, a clear indicator of the importance of DOL as a system property.

Conclusions about the influence of environmental constraints, i.e. task availability and cost, appear mixed. For linear-threshold models, task costs increased DOL at the 1:1 and 3:1 ratios when a switch from resource-rich/low-task cost to resource-constrained/high-task cost occurred. In scenarios without a switch, DOL increased almost linearly up to the 5:1 ratio. However, LTM-based colonies produced more pins in constrained environments and evolved LTM-based colonies produced more pins on average once a dynamic change in environment occurred. FTM-based colonies were more productive in environments with a low-task cost, even out-pacing LTM-based colonies in certain 5:1 population to task ratios. FTM-based colonies; however, appeared unable to return to pre-production rates once a bifurcation occurred; leading to the conclusion that LTM-based systems are more resilient. An unexpected result due to the high-use of the FTM in research and worthy of further investigation.

Furthermore, contrary to previous results [33], the data showed that high specialization, i.e. high DOL, is not required for high work output. This finding has two main results. One, agent decision functions create different levels of division of labor which is not necessarily correlated to resource yields, leading to; two, since no correlation appears to exist, then new agent decision functions need to be created and tested. The linear threshold model presented in this work showed superior resource production and resiliency to the well-researched fixed threshold model in spite of lower division of labor scores. In Adam Smith's Pin Factory domain, results support choosing the linear threshold model over the fixed threshold model in most environments; leading to the conclusion that it is possible

that current agent decision functions are possibly inadequate in many domains. Clearly, future work needs to both develop, and test, various agent decision functions in multiple domains to support, or refute, such a conclusion.

6.7 Conclusion

The work presented here studied the impact of agent decision functions, population size, and environmental constraints on a system's division of labor. Two agent decision functions, the Fixed-Threshold and Linear-Threshold Models, were implemented and tested in Adam Smith's Pin Factory domain. Experiments showed that agent decision functions and population size control DOL scores within boundaries related to environmental constraints.

Results showed that the FTM excelled in resource-rich, low-task cost environments, while the LTM performed better in resource-constrained, high-task cost environments. Additionally, the LTM outproduced the FTM in a majority of scenarios with the exception that FTM exhibited maximum-efficiency values with larger population-to-task ratios. The latter result has been shown in some biological work, and is a new finding for engineered systems. Finally, LTM-based colonies were more resilient to dynamic environmental change. These results indicate that further research should focus on agent decision models.

Aside from the differences in agent decision function behaviors, population size was found to have a positive correlation with DOL scores and system resource production. Both DOL scores and resource production increased as the population increased, regardless of the choice of an agent decision function. However, population size did not appear to affect system resiliency.

Finally, DOL scores cannot predict resource output outcomes for an engineered system. The statistically significant differences in resource production for the FTM and LTM-based colonies clearly indicate that higher DOL scores are not indicative of higher resource yields and, by extension, labor efficiency. Leading to the conclusion that higher DOL scores, i.e.

higher agent specialization, does not necessarily lead to higher resource output; an observation contrary to findings from previous work [33]. LTM-based colonies held lower DOL scores than FTM-based ones across all categories, but consistently outperformed FTM-colonies in the majority of the experiments. These results again highlight the critical role an agent's decision function plays in system performance.

The results from this chapter further support the overall hypothesis that agent speciation creates robust and adaptable systems. Furthermore, for the first time in this work, results empirically show that entropy-based agents are more robust to environmental change than non-entropy based agents with the same capabilities in the same dynamic environment. However, there are indications that certain environments, such as those that are resource rich, may favor less speciated populations when the desired end state is high resource yield. This shows that the environment also influences macro-level outcomes and must be taken into account when designing multi-agent systems.

VII. A DEVS-Based Approach Towards Hierarchical Ecosystem Modeling

7.1 Introduction

This chapter presents a formal modeling framework for engineering complex adaptive systems.¹ Complex systems are difficult to model because interactions between their internal components and the environment can create unpredictable behaviors. However, these unpredictable behaviors, known as emergent behaviors, can be beneficial as they make complex systems adaptable. The challenge is to design and model complex systems in a manner that allows the system to be highly adaptable, i.e. possess emergent properties, but still function as designed. The ecosystem-based modeling framework presented here maps ideas and terminology from emergence research, such as niches and signals, to discrete event simulation constructs. By defining niches as coupled models, and restricting the signals and resources that flow into, through, and out of them, one creates a highly adaptive complex system. Experiments using the framework to model two competing ant colonies show how the framework can construct a highly adaptive system while capturing root causes of emergent behaviors such as predator-prey oscillations and agent hunting. The novelty of the approach comes from its linking of emergence concepts to a discrete-event simulation framework.

7.2 Motivation

Engineered complex adaptive systems (CAS) are being fielded and interconnected with increasing frequency. Like their natural counterparts, these systems can display unpredictable behaviors that are difficult to identify, predict, and control [25]. These emergent behaviors [55, 14] create challenging domains of study. From an Artificial Intelligence

¹The majority of this chapter was submitted to the *Simulation: Transactions of Modeling and Simulation International Journal*, Apr 2019, for publication consideration.

(AI) perspective, these systems provide difficult prediction and control environments as these domains teem with subtle agent interactions and imperfect information [101]. However, these emergent behaviors in complex natural systems create robust and adaptable systems; desired traits for engineered systems working in dynamic environments ranging from local city traffic [20] to the asteroid belt [113]. The challenge is to create complex systems that can leverage emergent behaviors, granting system adaptability and robustness while providing mechanisms to identify and control those system specific behaviors.

Many researchers [13, 55, 57, 93, 96, 103, 108] propose that complex behaviors in natural systems emerge from local agent interactions at multiple system levels. At the center of these interactions lies the transfer of information through agent signaling and evolution. Agent signaling is the direct, or indirect, sharing of information between agents or their environment. For example, bee and ant colonies continuously share information via waggle dances [107] and pheromone trails [137]. Information sharing creates organized group behaviors, such as food retrieval parties [13] and armies amassed for nest defense [137]. These behaviors are a direct result of Darwinian evolution where complex behaviors evolved with colonies adopting successful strategies while abandoning failed ones [26]. For an engineered CAS to be successful, it needs to propagate information between agents and evolve as its environment changes. Information sharing and evolutionary mechanisms create the conditions for emergence and system survival. The primary obstacle for building such systems is how to model them. A proposed model must be able to incorporate known common traits found in natural systems: hierarchical structures, information sharing between those structures, environmental stimuli, decentralized execution of tasks and adaptation.

To meet this challenge, this chapter presents the ecosystem DEVS-based framework (eDEVS) which incorporates portions of multi-level DEVS (ml-DEVS) established by Uhrmacher et al. [131], dynamic structure DEVS (DSDEVS) by Barros [6] and Holland's

Echo model [54]. eDEVs considers each hierarchical level in the ecosystem as a coupled model, defined as a *niche*, that conducts and constrains the types of signals and information passed between them. Signal and resource passing between levels is handled through either dynamic message queues, or by modifying global variables stored in a separate atomic model. By logically separating the niches, the simulation is allowed to dynamically switch objects during runtime based upon predefined threshold events allowing for system adaptation. Using eDEVs, one can simulate complex behaviors based upon inter, and intra, niche signaling. Additionally, eDEVs provides for testing and analysis of temporal objects, such as mid and long-term system goals, by defining them as operational and strategic models, generalizing eDEVs' applicability.

The rest of the chapter is divided into five sections. Section 7.3 presents an overview of classic and ml-DEVs as well as an introduction to Holland's Echo model. Section 7.4 defines eDEVs and how it shores up some of the shortfalls in the ml-DEVs and Echo model approaches. Section 7.5 provides an example implementation of eDEVs with Section 7.6 showing how eDEVs can trace upward and downward causation behaviors. Section 7.7 concludes the paper.

7.3 Background

In his book, *Signals and Boundaries*[57], Holland finalized his holistic view of complex adaptive systems (CAS) which he started in his influential work, *Hidden Order*[54]. He argues that agent signaling and defined niches comprise the major components of a CAS. Holland defined niches as bounded areas where agents reside. Agents inside these niches communicate directly, or indirectly, through signals. Only specific signals can traverse the niche boundaries and impact the decisions of agents residing in other niches. These inter-niche signals tie niches together into an ecosystem. Aside from signals, resources may also flow from one niche to the next through environmental change, such as a human

dropping a piece of fruit by an ant hill or a sudden rain burst destroying ant pheromone trails. Both events provide external stimuli to the ecosystem and impact agent decisions. The interplay between intra and inter-niche signaling and resource exchanges create a dynamic ecosystem where each niche relies upon signals and resources from others to survive. Holland's rainforest example provides a concise representation of a highly dynamic, intertwined ecosystem rich in inter and intra niche signaling and resource recirculation. Each niche, higher up the topographical levels of the rainforest, not only rely on lower levels for resources but provide some benefit to the levels they build upon, creating a symbiotic relationship. Changes at any level will result in larger changes across the system.

Using Holland's proposed view, the environment and any other systems a CAS may be coupled with, influence system behaviors. Additionally, the emergence of system behaviors also relies on internal agent interactions. An architecture must incorporate not only coupled hierarchies but the interchange of resources and signals between them to correctly model CAS behaviors, a view that is in line with the systems theory definition of complex systems. According to systems theory, a complex system is a set of interrelated components, bound by intricate relationships with one another, working together towards a common goal [79]. Engineered CASs exemplify this model as engineers create them to solve or help solve, a problem. Finally, the environment surrounding the system influences system behavior. In summary, to model a CAS, an architecture must create coupled hierarchies, provide a means for tracing resource and signal interchanges between micro and macro levels, allow for model interchangeability during runtime, and enable dynamic coupling between the system and its environment. The discrete event simulation (DEVS) formalism provides a promising starting point.

The classic DEVS formalism introduced by Zeigler [143] and its extended port model [134] uses a combination of defined atomic and coupled models to model complex systems. DEVS is attractive for those trying to model complex systems as it can build hierarchical

models by coupling atomic models together. Each atomic model possess a set of states, $S = \{s_1, s_2, \dots, s_n\}$ that the atomic model may assume. States are accompanied by transition functions, δ_{init} , that define which state may follow another. Furthermore, each state possesses an associated time, ta , that defines how long an atomic model stays in a particular state. In order to accommodate external inputs, one defines a set of accepted input ports, $X = \{x_1, x_2, \dots, x_n\}$, which can cause state changes defined in the set, δ_{ext} . Finally, output functions, λ , produce an output event from the set, $Y = \{o_1, o_2, \dots, o_n\}$, associated with an internal state change. Altogether, the tuple $\langle X, Y, S, q_{init}, \delta_{init}, \delta_{ext}, \lambda, ta \rangle$ defines an atomic model.

When one links atomic models together, one creates a coupled model. Model instances, D , contain the atomic models included in the coupled model. The set, M_i holds each model specification; the atomic model 8-tuple defined previously. To allow information passing, one defines model influences, I_i . However, no model may influence itself, and only links within the coupled model are allowed. Having only internal links precludes sharing state with, or taking input from, outside the coupled model. Therefore, the sets of events, X_{self}, Y_{self} , represent the events allowed into and out of the coupled model, producing the 5-tuple, $\langle X_{self}, Y_{self}, D, M_i, I_i \rangle$. One creates hierarchies of interacting models by coupling atomic models together and defining input and output events accordingly. One can find a thorough overview of classic DEVS in Tendeloo's work here [126].

Although it provides mechanisms for coupling models and building hierarchies, DEVS does not provide a way to model high-level dynamics, macro-level behaviors, or micro and macro level interactions [131], which are defining characteristics of complex adaptive systems, inspiring Uhrmacher *et al* to construct multi-level DEVS (ml-DEVS)[131]. In ml-DEVS, coupled models contain a state, a behavior, and additional input and output ports that connect micro and macro models and enable information tracking between both models [131]. As information moves in both directions, one can trace upward and down-

ward causation effects as model states change at both levels. Additionally, these defined couplings create a means of keeping encapsulation as only certain internal variables are available to the coupled models [122].

Steiniger *et al*[122] extended ml-DEVS further by including a method for modeling environment-agent and agent-agent communication. Their model incorporates multi-coupling which defines multiple types of ports whose availability changes depending on the current model state. These ports restrict the information in and out of the model during the current timeframe. Additionally, their model captures the idea of sending information up and down the hierarchy to other models (i.e., micro and macro models) by strictly defining passable values. Capturing information flow is the most notable extension of the ml-DEVS model.

Some biological modelers [42] use these extended DEVS formalisms because they effectively model many of the same problems associated with modeling complex systems: linking micro-macro level attributes and dynamics [131], modeling environmental influences [122], and allowing dynamic switching of models during runtime [6]. However, the separation of micro and macro-DEVS models appears unnecessarily restrictive. For example, Steiniger *et al* state that one can layer micro-models on top of one another by considering them macro-models [122]. If they can all be regarded as macro-models then why use the micro-model at all? Furthermore, if each model is the same, then the parallel DEVS [126] approach can simplify the models even further.

In parallel DEVS, models pass information via each other's message queues. At each time step, the agent model parses its queue for information that may lead to state changes and enabling cyclic information flow [102]. Parallel DEVS holds similarities to the Layered Queueing Network (LQN) [134]. In an LQN, messages pass synchronously and asynchronously across modules. These messages create inputs to other modules in the simulation. An LQN can model multilayered software and hardware interfaces and can evaluate distributed systems [134].

Even with this simplification, one is still building multiple sets of coupled models and trying to enumerate the possible combinations of ports and states within the simulation. The problem space exponentially grows as one adds more agent and environmental models. DEVS also extracts away discrete time events which leads away from real-world ecosystem models. In the real world, where engineered CASs interact, the environment continually changes, meaning events fire at every time step creating conditions for rapid system behavior changes. One must account for both internal and external pressures (i.e., stimuli) necessitating a broader modeling approach.

If one frames a CAS and its environment as an ecosystem, one must combine the best features of multi-level, parallel, and adaptive DEVS architectures. The required ecosystem architecture must allow dynamic model change during simulation, create separable hierarchical levels, incorporate environment-agent and agent-agent information flow, and strictly enforce encapsulation between models. The next section defines the eco-DEVS (eDEVS) architecture which integrates these desired traits.

7.4 Eco-DEVS Architecture

This chapter's primary premise is that components inside natural and engineered CASs work towards multiple, temporal goals simultaneously. As one moves up the level hierarchy, temporal goals transform and extend. For example, agent goals are almost instantaneous; their decisions affect their immediate state. However, these micro-level decisions move the entire system towards the completion of higher-level goals. As a complex system may have many micro-levels, it is necessary to couple each level to others it may affect. To keep terminology precise, every distinct logical level, made up of defined components, is considered a niche. Each niche determines the signals and resources that may pass into, and out of, the elements it contains. All components within a niche can freely interact with one another through direct or indirect communication, i.e. agent signaling. Components

can also transmit and receive signals to other niches through a signal collection and transmission device, i.e., message passing, as allowed by its niche. Building a house serves as an illustrative example.

The desired macro-level outcome is a house which one breaks into sets of micro-level goals, e.g., plumbing, electricity, framing, roofing, etc. These micro-level goals possess associated tasks. Single agents, or small teams of agents, possess the skills necessary to accomplish simple tasks, such as nailing a piece of wood, pouring concrete, etc., that aggregate to the accomplishment of a micro-level goal. Over time, agents complete more and more micro-level goals, eventually meeting the desired macro-level outcome.

One can measure system status through these micro-level goals, answering important questions such as will the house be completed on time, or on budget? If a time constraint is failing, then one must dig deeper for causality, so one must establish a mechanism for tracing influences up and down the hierarchy.

For example, if the workers do not show up to work for a few days, then micro and macro-level timelines become endangered. The same applies to workers choosing not to complete specific tasks on time, or in a particular order, exemplifying upward causation as these agent level decisions naturally impact micro and macro-level outcomes. Furthermore, decisions made at the macro-level level flow downward through the system affecting micro and agent level decisions. If a home buyer wants to change the structure of the house, this necessitates a change in the micro and agent level goals. A macro-level change also incurs a cost, in resources and time, and must be captured by the model. More succinctly, lower levels present dynamic possibilities, scoping what a system can accomplish, while higher levels provide a purpose that lower levels work towards [3].

Using the DEVS formalism, the eDEVS framework outlines the interactions between atomic models, i.e., agent, micro and macro levels. As complex systems are hierarchical, this formalism creates strict feedback cycles in the framework where higher-level guidance

signals flow downward to the micro level and then, by extension, to the agent level. As agents accomplish tasks and meet micro-level goals, signals pass up the hierarchy to the macro-level. At the strategic level, thresholds form the basis for macro-level shifts, depending on how the signals change the perception of the world at the macro level. Thresholds create the basis for the firing of events which can impose state changes upon the system. In the eDEVS framework, one atomic model is used for all possible levels in a complex system, simplifying its implementation.

The atomic model for eDEVS is defined by the tuple $\langle id, tag, S, q_{init}, \sigma_{init}, X, Y, \lambda, V, R, M, ta \rangle$. Each atomic model contains a unique identifier, id , and an interaction genotype, tag , that describes interactions between models. A function, $f(state)$ (not shown), constitutes the state analysis function that an atomic model processes which executes the appropriate transition function λ upon completion, effectively representing the model's decision-making process. X and Y represent sets of input and output signals the model can generate. Inline with ml-DEVS, V describes the values that may be passed by, and to this model. The set R characterizes the resources an atomic model may either generate, gather and use from the local environment or other models. Similar to variables, the set R captures the recirculation of resource energy in natural systems [57]. The set M defines models this particular atomic model may couple to, and finally, ta represents the length of the decision cycle for the model. Once ta expires, the agent makes a decision.

A coupled model in eDEVS is referred to as a niche [57] and contains the set of atomic models that compose it, D , and the possible input and output signals, X and Y , to form the tuple $\langle X, Y, D \rangle$. The internal atomic models may have their own defined interactions and signals. The coupled model only worries about external signaling. Additionally, the models that compose the coupled model expresses its overall behavior. In other words, like DEVS, the coupled models do not have a defined behavior. Behaviors expressed by coupled models reflect the interactions of internal components and external stimuli. The

coupled models' primary purpose is to limit input and output signals, thereby limiting the stimuli internal agents receive from other coupled models as well as messages they can pass, creating a logical grouping of agents into a niche as described by Holland [57] where only specific signals may permeate niche boundaries and flow through the ecosystem.

Figure 33 presents a notional strategy guided three-tier system diagram using the eDEVS framework. Hierarchical niches create boundaries around each acting eDEVS component, e.g., tactical, operational and strategic. These niches restrict the flow of information into and out of the components they contain and may be viewed as DEVS coupled models. However, by their dynamic nature, niches require the ability to add and remove atomic models during runtime, similar to dynamic DEVS. Components may interact with each other inside their bounded niches; however, eDEVS strictly controls output signals. Signals from the tactical niche accumulate in the Message component which is read by the Operational niche. Once the Operational components finish reading Tactical signals, they relay information both up and down the hierarchy. Finally, the Strategic component assesses the current state of the Operational goals and how they relate to Strategic ones. The Strategy component can then update global variables, accessible by the Tactical niche components. The Tactical models receive signals from the Strategic, Operational and environmental levels. eDEVS provides traceable upward and downward causation signal pathways. As modules reside inside separate niches, they can be replaced by other modules easily without disrupting the simulation. In this manner, eDEVS provides the required conditions for modeling a CAS: dynamic model change during simulation, separable hierarchical levels, environment-agent and agent-agent information flow, and encapsulation.

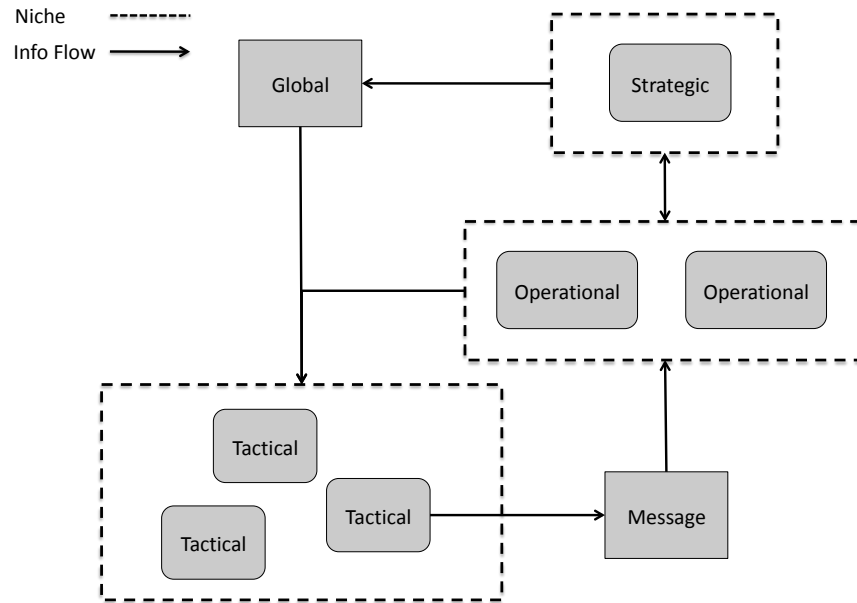


Figure 33. Notional system diagram using eDEVs structure. The dotted lines represent hierarchical niches with strict input and output signals specified by directional arrows. In this particular diagram, messages and global variables represent the message signaling techniques employed by niche subcomponents.

7.5 Example

Competition between two ant colonies in a restricted geographical area provides a working example of the eDEVs framework. Each colony selects a strategic vision in which to outmaneuver its competition which guides agent interactions from the system level and establishes operational goals the colony must meet to win. A colony wins by either eliminating the competition or reaching its operational goals.

At initialization, each colony randomly selects an initial strategy which may result in long-term competitive or cooperative behaviors simulating ant colony interactions found in the natural world[137]. In competing strategies, ants can either engage enemy ants, gather resources, or move to adjoining spaces. At the end of each time step, ants may yield resources to the colony, incrementally accomplishing operational goals.

The following defines an ant colony strategy (i.e. macro-level) model:

$$id = \{ColonyA, ColonyB\}$$

$$tag = \{none\}$$

$$S = \{Rock, Paper, Scissors\}$$

$$q_{init} = \{Rock\}$$

$$\sigma_{init} = \{Paper \rightarrow Rock, Paper \rightarrow Scissors,$$

$$Rock \rightarrow Scissors, Rock \rightarrow Paper,$$

$$Scissors \rightarrow Rock, Scissors \rightarrow Paper\}$$

$$X = \{stimuli_{niche}\}$$

$$Y = \{updateGlobal, message\}$$

$$\lambda = \{increase(stim_{engage}) \rightarrow global,$$

$$increase(stim_{gather}) \rightarrow global,$$

$$increase(stim_{replicate}) \rightarrow global,$$

$$changeOps \rightarrow Ops\}$$

$$V = \{0, 10\}$$

$$R = \{none\}$$

$$M = \{atomic, niche\}$$

$$ta = \{10\}$$

The colony strategy set agent genotypes. The first letter of every agent genotype matches the colony strategy. The colony assesses the success of the overall strategy every ten time steps. If the strategy is working, it can increase the stimuli for agent engagement (i.e. aggression), resource gathering, or agent replication. Otherwise, if a strategic shift is necessary, then the strategy changes and raises signals to both the Operational and Tactical levels. The Operational goals are reset and all agents at the Tactical level change the first letter of their genome accordingly. In this manner, the colony adapts to changes in colony population. Again, highlighting the interplay between macro and micro levels as the micro-levels limit what can be accomplished through their abilities and influence the macro-state while the macro-level dictates agent purpose.

At an operational level (i.e. micro-level), ants must accomplish multiple concurrent goals. How quickly ants fulfill these goals impacts Operational signaling. After five time steps, if ants provide a specified percentage of resources, then the Operational level will not signal any changes. Otherwise, the Operational level will increase a stimulus value associated with a tribute task that ants can accomplish. The following DEVS model shows an ant colony at the operational level.

$$id = \{RockOpsA, PaperOpsA, ScissorOpsA,$$

$$RockOpsB, PaperOpsB, ScissorOpsB\}$$

$$tag = \{none\}$$

$$S = \{RockOps, PaperOps, ScissorOps\}$$

$$q_{init} = \{RockOps\}$$

$$\sigma_{init} = \{RockOps \rightarrow ScissorOps,$$

$$ScissorOps \rightarrow PaperOps$$

$$PaperOps \rightarrow RockOps\}$$

$$X = \{stimuli_{niche}\}$$

$$Y = \{updateGlobal, message\}$$

$$\lambda = \{increase(stim_{tribute}) \rightarrow global,$$

$$increase(stim_{gather}) \rightarrow global,$$

$$increase(stim_{engage}) \rightarrow global$$

$$totalPercentage \rightarrow Strategy$$

$$antsAlive \rightarrow Strategy\}$$

$$V = \{0, 1000\}$$

$$R = \{R, P, S\}$$

$$M = \{atomic, niche\}$$

$$ta = \{5\}$$

Finally, at the tactical, or agent level, agents choose actions based on the variable response method described in Bonabeau [13]. Agents signal both other agents and the operational level but in two different ways. The absorption of resources, agent engagement, and movement all produce stigmergic signals while, at the end of each sense-decide-act loop, agents send direct signals to the operational level. For our experiment, agents filled a message queue that the operational entities then processed before the next simulation iteration. Additionally, at each time step, each resource grid randomly produces upwards of three new resources for agents to gather.

The threshold equation (see Eq: 22) from Bonabeau [13] necessitates internal and external thresholds for each task. In our simulations, a global module, accessible by tactical modules (i.e., agents), held the external thresholds s_n . Internal thresholds θ_n were randomly initialized between (0,10).

$$T(s_n) = \frac{s_n^2}{s_n^2 + \theta_n^2} \quad (22)$$

Agents select the task with the highest value calculated via Equation 22. If the agent could not accomplish the selected task, e.g., no resources resided in the current square for collection, then the agent randomly picked another task to complete. A process that continued until the agent could accomplish some task. For these experiments, agents could always execute the movement task.

$$id = \{uniquestring\}$$

$$tag = \{genomstring\}$$

$$S = \{move, gather, engage, replicate, death\}$$

$$q_{init} = \{gather\}$$

$$\sigma_{init} = \{move \rightarrow gather, gather \rightarrow move,$$

$$move \rightarrow engage, engage \rightarrow move,$$

$$move \rightarrow replicate, replicate \rightarrow move,$$

$$engage \rightarrow replicate, replicate \rightarrow engage,$$

$$engage \rightarrow death\}$$

$$X = \{stimuli_{environ}, stimuli_{niche}\}$$

$$Y = \{message\}$$

$$\lambda = \{tribute \rightarrow ops, dead \rightarrow remove, task \rightarrow ops\}$$

$$V = \{R, P, S, alive, N, E, W, S\}$$

$$R = \{R, P, S\}$$

$$M = \{atomic, niche\}$$

$$ta = \{1\}$$

Figure 34 provides an object-oriented view of agent objects. The id tag is unique for each tactical agent, while the tag, i.e., genotype, may be identical to other agents. Three character strings composed of resources, $R \in \{R, P, S\}$, represent the agent tag. The first character in the tag aligns with the colony's strategy while the second two letters are drawn randomly from the set R . For engagement, the composition of the tag determines the winner. The regular rules of *Rock, Paper, Scissors* apply where Rock beats Scissors, Scissors beats Paper, and Paper beats Rock. After comparing the letters in the tags, the agent with the most wins survives the engagement and absorbs the resource pool of the losing agent. Nothing occurs when agents tie. If an agent dies during an engagement, the simulation removes it from the population and updates the agent counts for each colony.

Agents may move North, South, East or West, to adjoining squares. Diagonal moves are not allowed. Inline with Holland's Echo model [55, 57], agents may replicate but can only do so when they acquire resources equal to their tag. For example, an agent with tag $\{PPR\}$ cannot replicate until its reservoir holds at least two paper and one rock resources. If the agent chooses to replicate, the simulation removes these resources from the agent's reservoir and places an identical agent, with a unique ID, in the same square. Finally, at the end of an agent's sense-decide-act cycle, it may send a resource, via signaling, to the operational niche. The parameter θ models agent selfishness and is a probability drawn

randomly from a normal distribution [0,10] upon agent creation. During an agent's turn, it can only select one action with tribute being a free action executed at the end of an agent's turn.

Tactical Agent	Operations Agent	Strategy Agent			
id : String tag : String colony : String currentTask : int resources : String location : <int X, int Y> thetas : double[] alive : boolean	id : String goals : int[] resources : int[] completed : double void readUpdate(Message) double getUpdate() void changeGoals(char) void update(double [])	id : String numAgents : int completed : double void readUpdate(Message) void changeStrat() void calcUpdate() void update(double [])			
void decide(double []) String tribute() void gather() void engage() void replicate() void move() double threshold(int task) updateTag(char) Message createMsg()	<table><tr><th>Message</th></tr><tr><td>agentID : string colony : string string : tribute boolean : alive location : <int x, int y></td></tr><tr><td>void create(agentInfo)</td></tr></table>	Message	agentID : string colony : string string : tribute boolean : alive location : <int x, int y>	void create(agentInfo)	
Message					
agentID : string colony : string string : tribute boolean : alive location : <int x, int y>					
void create(agentInfo)					

Figure 34. Object-oriented view of agent models.

7.6 Experiment

Two colonies, A and B, were set into the grid world described in the previous section. Initial strategies for both colonies were randomly selected. The purpose of our experiment was to show how agent interactions at the tactical level create strategic change. Based upon the environment rules, one expects to see a cyclic shift in strategy between both colonies, e.g., if Colony B chooses *Paper*, and Colony A chooses *Rock*; eventually, Colony A will switch to *Scissors*, prompting Colony B to switch later to *Rock*. Colonies could only change strategies once every 20-time steps to avoid fast oscillations between strategies. Ten

thousand random initializations were run to test the ecosystem DEVS based framework.

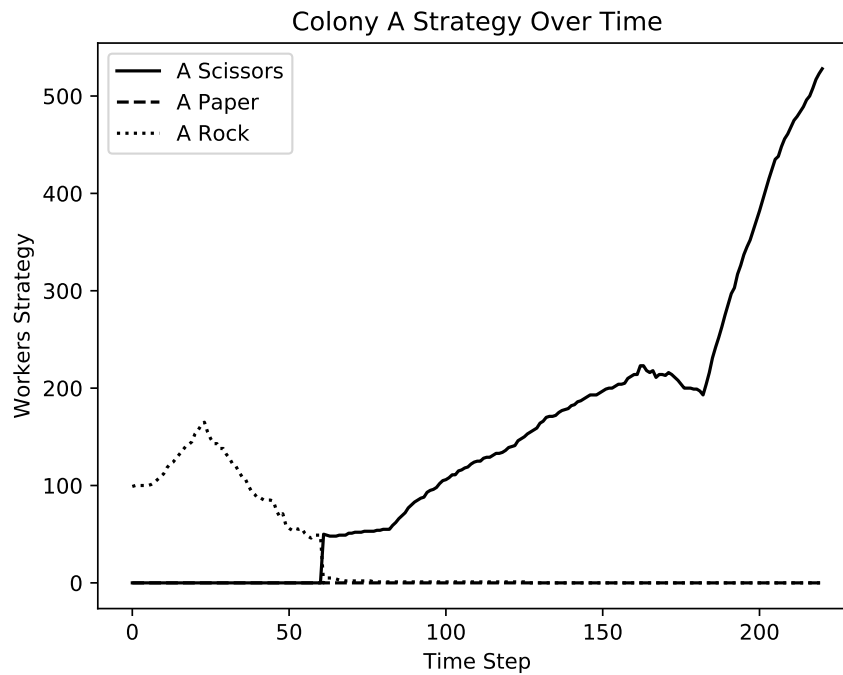


Figure 35. Colony A strategies over time. Colony A begins with the superior ‘Rock’ strategy and experiences an increase in agent population. However, the population begins to decline with Colony B switches to the ‘Paper’ strategy. Eventually, Colony A chooses the ‘Scissors’ strategy, eventually eliminating Colony B’s population. Here one sees the dynamics between macro-level guidance and micro-level decisions as Colony A’s strategic changes influenced agent-level interactions, resulting in a positive outcome.

Figures 35 through 37 show a simulation outcome when colonies started with opposing strategies. Although a relatively simple simulation, complex behavior patterns such as predator-prey oscillations predicted by Lotka-Volterra equations [69] emerged. Importantly, one can trace why the oscillations occurred when they did. For example, by tracing the current strategies of each agent over time, one sees the rise and fall of colony populations as the colonies change strategy. By overlaying the data in the three figures, one sees rises and falls in colony populations soon after a change in strategy. The only time a colony did not successfully change was after time step 100. At time step 100, both colonies held stable populations encouraging both to stay with their current strategic choices. During this time, Colony A’s population begins to increase, eventually affecting Colony B’s popu-

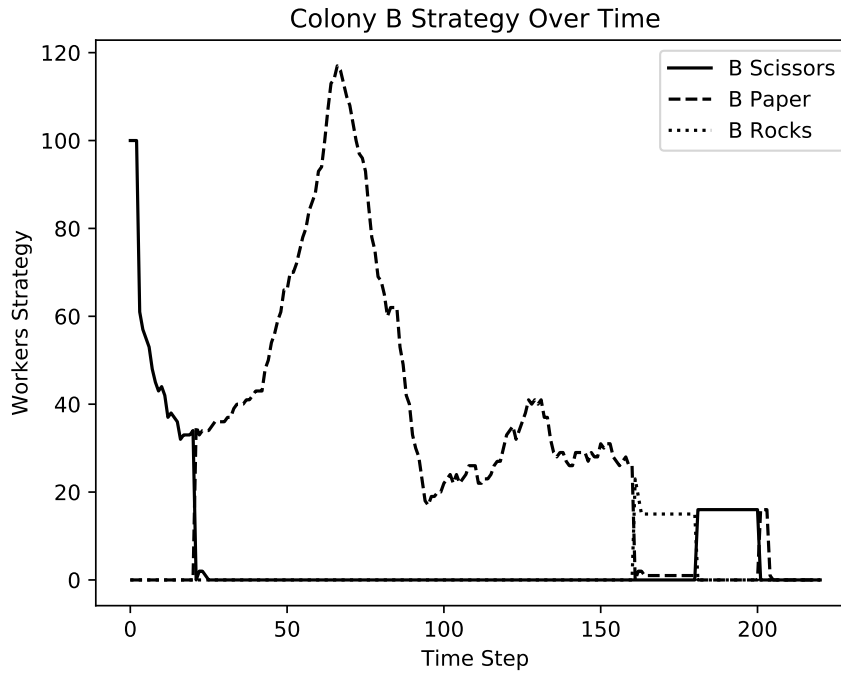


Figure 36. Colony B strategies over time. Although Colony B starts the simulation with a disadvantageous strategy, it switches to a superior at time step 20. Colony B then experiences a population explosion, almost leading to the destruction of Colony A. However, it experiences a population decline when Colony A switches strategies. Interestingly, although Colony B holds the inferior strategy, it can maintain its population. In response to Colony A’s population increase, it changes to the superior strategy of ‘Rock’ and begins to impact Colony A’s population but only for a short time. As the strategy does not increase its overall population, Colony B begins oscillating strategies at time step 140, only to succumb when it chooses the inferior strategy ‘Paper’ when Colony A holds a distinct population advantage and strategy.

lation. Colony A’s pressure on Colony B begins an oscillatory strategy selection behavior at time step 140 (Fig 36). As each strategy fails to increase its overall population, Colony B works through different strategies every 20-time steps. Eventually, the combination of a poor strategy, ‘Paper’ when Colony A has ‘Scissors,’ and Colony A’s superior numbers, destroys Colony B.

Here one must note that the Lotka-Volterra equations predict predator-prey populations in a pristine environment, where birth and death rates remain stable. In this simulation, neither rate is steady over any period of time as birth rates and agent deaths rely on resource availability, a combination of internal and external stimuli, and the colony’s chosen strat-

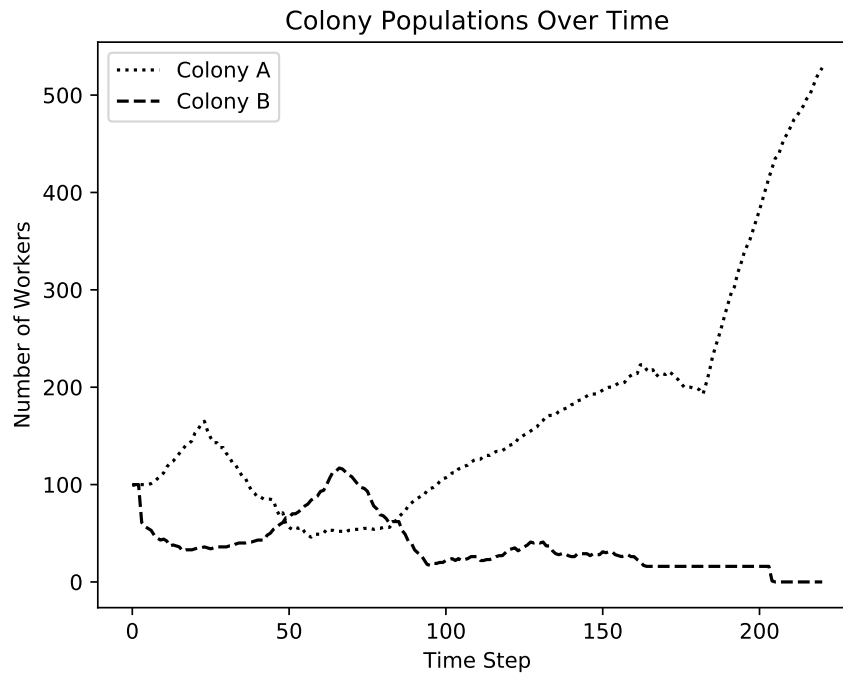


Figure 37. Colony populations overtime for competing strategies. Clear predator-prey oscillation patterns emerge as both colonies vie for supremacy in their environment. The 90-degree offsets for the populations mirror the expected predator-prey oscillations predicted by Lotka-Volterra equations, showing how the eDEVs framework can capture emergent behaviors in complex adaptive systems.

egy. Additionally, the agents can prey upon one another, creating competition between two apex predators in the environment. Although resources are plentiful, using only population counts as a guide for strategy ultimately leads to Colony B's destruction.

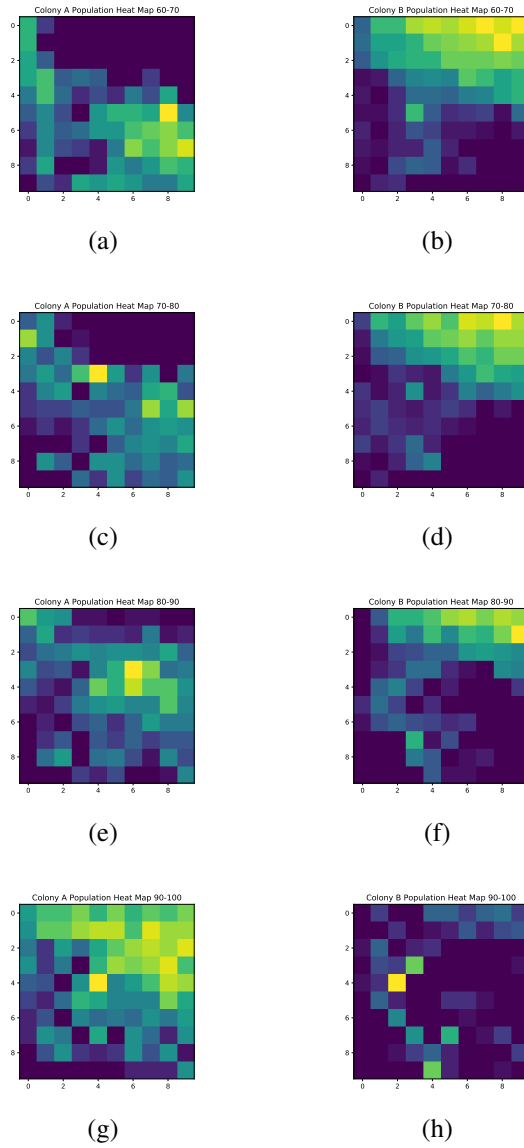


Figure 38. Colony A and B population transitions from time step 60 to 70. Lighter squares represent higher population densities. Although not implemented in the system, a hunting behavior emerged as a clear movement towards Colony B's population occurs when Colony A changes to the 'Scissors' strategy at time step 40 and agents receive an increase in their aggressive stimuli from the macro-level.

In addition to predator-prey oscillations, a hunting strategy emerges when Colony A assumes a new strategy at time step 60 (see Figure 38). Armed with the superior 'Scissors' strategy combined with an increase in engagement stimulus from the macro-level, Colony A agents begin migrating towards locations containing Colony B agents. The hunting behavior was not coded into the system and is likely the result of a lack of resources in the

original Colony A positions, pushing the population towards Colony B. Colony B agents then move away from Colony A agents as nodes become overpopulated.

It is clear from the experiment results that the eDEVS framework adequately modeled a multi-tiered hierarchical system with one atomic model while incorporating environmental inputs and strict inter-level signal and resource handling while allowing decentralized execution of tasks at each independent level as well as agent adaptation meeting the requirements laid out for such a framework in Section 3.1. Additionally, by capturing agent and system stimuli, agent locations plus resource allocations, one could trace causal influences over time responsible for the emergence of predator-prey oscillations and hunting behaviors, indicating that the eDEVS framework allows one to engineer emergent systems that can produce emergent phenomenon, and more importantly, allow for causal analysis when those behaviors occur.

7.7 Conclusion

This chapter presented the eDEVS framework and showed how one can use it to model hierarchical systems, as well as track micro-macro and macro-micro level interactions, generating behavior causation linkages missing in the majority of previous formal modeling approaches. Experimental results showed how the framework could capture emergent, complex natural dynamics, such as predator-prey oscillations and hunting behaviors, through the use of strategic guidance based on internal agent feedback. The framework provided causal chains to allow logical tracing of how these behaviors emerged. Furthermore, the experiment showed how causality of emergent behaviors is not based solely upon upward or downward signals but is a combination of both.

Additionally, the experiment showed how with minor strategic and operational guidance, agents at the tactical level were able to accomplish complex tasks as well as show adaptability once strategic goals changed. The eDEVS framework can produce adaptable

and robust systems based upon agent-strategy signaling. Overall, results showed how the framework could be used to test various high-level strategies and see what system dynamics they produce. One can also leverage the framework to test opposing strategies and how long one could expect to realize the benefits of strategy change or how much strategy change impacts the system under test. Finally, eDEVs captured environmental input and how it affects agent performance and, by extension, the strategy of a system.

VIII. The Rise of Order: Emergence and Speciation in Multi-Agent Systems

8.1 Introduction

Emergence in natural systems is the development of complex behaviors that result from the aggregation of simple agent-to-agent and agent-to-environment interactions.¹ Emergence research intersects with many disciplines such as physics, biology, and ecology and provides a theoretical framework for investigating how order appears to spontaneously arise in complex adaptive systems. In biological systems, emergent behaviors allow simple agents to collectively accomplish multiple tasks in highly dynamic environments; ensuring system survival. These systems all display similar properties: self-organized hierarchies, robustness, adaptability, and decentralized task execution. However, current algorithmic approaches merely present theoretical models for emergent systems without showing how these models actually create hierarchical, emergent systems. To fill this research gap, this chapter presents a conceptual algorithmic approach based on speciation - defined as morphological or physiological differences in a population - that results in hierarchical emergent phenomena in multi-agent systems.

Results show that speciation creates system hierarchies composed of goal-aligned entities, i.e. *niches*. As niche actions aggregate into more complex behaviors, more levels emerge within the system hierarchy, eventually resulting in a system that can meet multiple tasks and is robust to environmental changes. Speciation provides a powerful tool for creating goal-aligned, decentralized systems that are inherently robust and adaptable, meeting the scalability demands of current, multi-agent system design. Results in base defense, *k-n* assignment, division of labor and resource competition experiments, empirically show that speciated populations create hierarchical self-organized systems, capable of meeting

¹The majority of this chapter will be submitted to the *Emergence: Complexity and Organization* journal for publication consideration.

multiple tasks, and are more robust to environmental change than non-specified ones.

8.2 Motivation

In today's world, the fielding of multi-agent systems composed of autonomous entities is rapidly increasing [25, 49, 68]. These systems must be robust to agent loss, adaptable to environmental change, and possess the ability to concurrently complete multiple tasks. The high complexity of agent interactions, possible system states, and volume of agent communication and sensor data overwhelms centralized control approaches [17, 91, 129]. Additionally, like their natural counterparts, these systems contain non-linear behaviors that are difficult to predict and control [25, 49]. These behaviors, referred to as emergent [14, 55], create a challenging design problem.

The problem facing engineers is how to design a complex, multi-agent system that is inherently decentralized yet adaptable and robust. One way to do this is by developing a self-organizing system that has an emergent property matching a system goal, i.e. a desired macro-level behavior. However, an agreed upon framework, or model, for engineering such systems currently eludes the field [69, 57, 30]. Emergence researchers contend that emergent systems possess beneficial characteristics such as self-organizing behaviors, robustness, adaptability, and decentralized control [13, 55, 57, 93, 96, 103, 108]; all desired characteristics for engineered systems. While many modeling approaches exist [57, 103, 49], relatively few [33] look at the impact of agent physiology or morphology upon emergent behaviors. This work formulates multi-agent, emergent systems based upon one major concept: speciation, and studies its role in the evolution of weak emergent phenomena to achieve a macro-level goal.

Speciation is the diversity of agents in a population. Differences can be *morphological* or *physiological*. Morphological differences are physical differences such as larger body size and mandibles that differentiate soldier and worker castes in some ant colonies

[139]. These physical differences imply forced speciation through long-term evolutionary processes. For engineered systems, human-derived differences such as various sensor suites and engines replace natural evolution processes with human-led ones. The development of different types of vehicles and refinement of their capabilities is still evolutionarily iterative, just artificial.

Physiological differences refer to how the agent functions, and changes based on how the agent digests information from its environment [4]. For example, the social spider *Anelosimus studiosus* all look the same, and all have the same sets of skills, e.g. web maintenance, hunting, brood care, etc. However, Holbrook, et al. [53] found that docile and passive members of this particular species engage in particular tasks aligned with temperament, with aggressive spiders trending towards hunting while passive ones took on web maintenance and brood care tasks. These differences lie within the spiders, and could be based on genetic material or learned experiences. The end result is physiological speciation. One primary trait of physiological speciation of interest here is the ability of an agent to change its function based on the current environment or learned experiences. This makes physiological speciation especially powerful as it creates agents with plasticity.

Biologically speaking, plasticity is where an agent adapts, either morphologically or physiologically, to changes in its environment [5]. Delving further into the habits of the social spider *Anelosimus studiosus*, Holbrook, et al. [53] found that if they founded a colony of all aggressive, or all passive, spiders, the colonies self-organized into groups that carried out all the required tasks, e.g. web maintenance, hunting, etc., regardless of the spider's internal bias. In other words, the spiders showed physiological plasticity by accomplishing less desirable tasks, ensuring colony survival.

This work hypothesizes that the combination of speciation with the beneficial characteristics of emergent systems, results in a generalizable, decentralized-control methodology for complex, multi-agent systems. At the agent level, speciation aligns agents into func-

tional groups, defined as *niches*. Each niche aligns towards a sub-goal, creating a hierarchical level composed of different niche *species*. As niche groups work towards completing sub-goals, their actions aggregate into an emergent macro-level behavior that matches a system goal. By using speciation, one can engineer a multi-level, hierarchical system whose inherent emergent behaviors support the system's overall purpose.

8.3 Background

Emergence, and the debate surrounding it, dates back to the late 1800s when G.H. Lewes noted certain natural phenomena could not be expressed in quantitative terms [24]. Lewes was attempting to explain how complex behaviors, such as birds flocking, could not be directly correlated to the sum of its parts. In other words, if one understood a bird and all its potential actions, e.g. hopping, chirping, flying etc., one would not guess that dozens of flying birds could self-organize into a cohesive flock that flew at the same speed, in the same general direction, without running into one another and without a leader coordinating their actions. Lewes termed these types of complex behaviors *emergent*. Emergence fell out of favor as breakthroughs in genetics made emergent theories about evolutionary behaviors inadequate as genes began explaining parts of human physiology [24]. However, in the late 1960s, emergence research reignited when Roger Sperry [24] coined the term *downward causation* to describe the functioning of the human brain, gaining momentum as Minsky [92] explored his theory of mind.

Broadly speaking, emergent behaviors are novel patterns displayed by complex systems. For example, a simple ant cannot find the shortest path between a food source and the nest. However, a million ants can construct such a path. This complex behavior, shortest path building, is displayed by the complex system, i.e. the ant colony, but not the individual ant. This line of thought results in two ideas: complex systems are composed of hierarchical levels with independent properties, and micro-level actions result in macro-level behaviors.

The latter idea infers possible causation arguments, i.e. certain micro-level actions should create specific macro-level behaviors, while the former implies some self-organizing power inherent to natural, complex systems.

Bedau defines two forms of emergent behavior: weak and strong [9]. Weak emergence is a behavior that can be reduced to actions and behaviors of agents at a lower level. Furthermore, weak emergent systems display upward causation tendencies where micro-behaviors effect macro-behaviors [9]. Strong emergence, on the other hand, is irreducible. Some [9, 28] believe human consciousness is an example of *strong* emergence as it is not deducible by current knowledge [28]. As such philosophical debate is beyond the scope of this dissertation, work instead focuses on proposed *weak* emergent behaviors as they are hypothetically reducible, and therefore, one is able to reproduce them with simulated models.

One primary advocate of emergence research is John Holland. His works, *Hidden Order* [54], *Emergence: From Chaos to Order* [55], and *Signals and Boundaries* [57], provide theoretical models and ideas on how emergence occurs. Holland proposes that emergent patterns rely on the atomic actions of agents as well as the behaviors (or interactions) between them [55] and models for such agents need to contain both (see Figure 1). Additionally, Holland states that the environment plays a role in the development of agent behaviors and interactions. The environmental argument is by Dautenhahn [27] in work about socially intelligent agents. Specifically stating that “intelligence can only be studied with a complete system embedded and coupled to its environment” and such agents must be adapted to their environment.

Corning extends both arguments with work on *synergy*. According to Corning, emergent behavior is the result of agents within the system, as well as the environment, aligning in *purposeful* or *synergistic* ways, producing both expected and novel behaviors [24]. Novel behaviors arise as agents react to environmental stimuli *at every level* of the system.

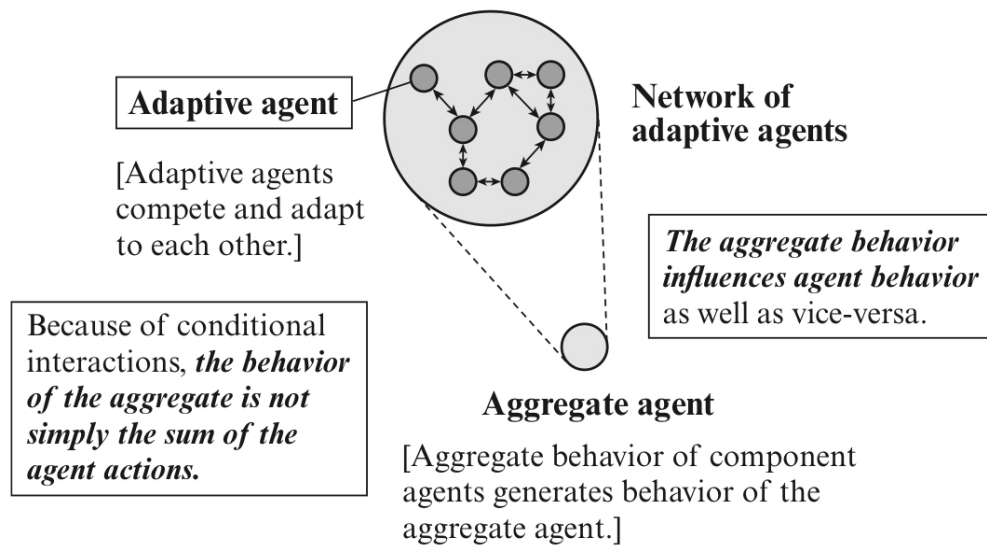


Figure 39. Complex Adaptive Systems [57].

A complex system should display emergent behavior if, and only if, there exist synergies between agents at various levels. Additionally, the environment in which the agents *live* guides agents with continuous feedback. Feedback enables the agents, and by extension the system, to learn and adapt.

Emergence research theorizes the following about complex systems. One, emergent systems consist of nested hierarchies built upon agent actions and interactions. Arguments for and against downward causation and exclusivity aside [9, 72, 61, 24], there is a prevailing belief that a hierarchical system relies upon macro-level behaviors emerging from micro-level interactions [13, 55, 57, 93, 96, 103, 108]. As one moves up the hierarchy, the behaviors encountered grow more complex, eventually forming a macro-level behavior that exceeds the capabilities of individual entities in the system [108]. Two, the building complexity of the system makes it more flexible, and therefore robust to environmental change [7]. Flexibility hinges upon the notion of, three, internal feedback loops. Macro-level behaviors influence and rely on changing micro-behaviors [55]. This view imposes a hierarchy of behaviors on the system a where cyclic feedback loops allow the micro to in-

fluence the macro which, in turn, influences the micro, possibly creating a new macro-level pattern, or extending or ceasing an old one [57].

8.4 Approaches

There exists a high interest in predicting and controlling emergent behavior in many fields ranging from biology and ecology to systems engineering and computer science, with a particular focus on mapping micro to macro-agent level behaviors. The majority of research centers on desired end-state prediction instead of studying the impact of micro-level interactions. For example, de Wolf, et al. [142] avoided tying micro-interactions to macro-behaviors by building a goal driven system with little concern for causality. In other words, they designed the system from the top-down, versus the bottom-up. Morandini, et al. [96] used a similar approach, again building a goal driven system; however, they allowed agent adaptation to build a more robust system. Morandini, et al.'s method acknowledged the link between micro and macro-agent behaviors, but, like de Wolf, et al., they avoided investigating these causal links. These approaches assume a reductionist view of micro and macro-agent level behaviors focusing on downward causation, i.e. macro-level goals guide micro-level interactions, ignoring benefits of tying micro-level interactions to macro-level goal accomplishment. Only Breddin [15] seems to advocate a systematic, top-down methodology where one studies a macro-level behavior and begins mapping out the components that lead to it's emergence. Their method at least acknowledges the role of micro-level interactions.

Pais [103] and Gignoux, et al. [41] used graph theory to map micro to macro-agent level interactions. Pais [103] used replicator-mutator dynamics (RMD) to analyze how some behaviors become dominant in social networks. Using a set of differential equations, RMDs model various strategies employed by agents inside the population and provide a method for offspring to inherit "fit" strategies as well as allow for mutations as seen in

many genetic algorithms [62]. RMDs provide a reasonable method for mapping local-interactions to a global state (i.e. strategy). However, it is unclear how scalable the solution is for systems that possess more than three global strategies.² Their work also concentrated on known global strategies found in nature, knowing the constituent parts *a priori*.

Gignoux, et al. [41] presented a graph theory based model and ontology to study the interactions between micro and macro-agent level behaviors. They modeled agents as nodes in a network with relationships defining the edges between them. In this manner, one can construct a visual topography of the system. The authors admit one major drawback to their model is the maintenance of the edge relationships. Adding and deleting edges is cumbersome and has the potential for erroneous links remaining in the system or even accidentally removing beneficial links [41]. Their network based approach arises from both Holland [57] and Mitchell [93] who proposed that network theory could play a role in understanding how complex systems evolve.

A common theme for these approaches is the use of simulation as a substitute for formal mathematical proofs. Relying on the law of large numbers, running a large number of simulations guarantees convergence to an average global state. As an example, Bonabeau, et al. [13] used Monte-Carlo simulations to produce close approximations of army ant raid patterns. Pais [103] and de Wolf [142] used simulation to verify their respective hypotheses with respect to global outcomes. By extension, one could expect that simulations could produce expected local outcomes (i.e. emergent behaviors below the macro-agent level) as well as global ones. The key issue is mapping agent and environmental interactions to those outcomes.

²Pais used a triangular simplex to graphically show the system shifting to one favored global strategy over two others [103].

8.5 Engineering Systems with Speciation

Speciation is morphological or physiological differences in a population and this work hypothesizes that it is key to designing robust and adaptable systems. This idea is supported, although not explicitly, by emergence researchers such as Wimsatt, Holland and Corning. Wimsatt believes that emergence involves an ordering, or organization, or various system components [140]. Holland acknowledges the role of speciation in creating diverse, natural ecosystems, where differences in species create cycles of resource recirculation, harvesting and sharing [57]. Speciation provides a mechanism where *meaningful* [24] function selection, e.g. sub-goal or goal alignment, creates self-organized *niches* that build a hierarchical system replete with internal feedback loops and emergent, macro-level behaviors.

Figures 40 and 41 provide a visual representation of our proposed system model that revises Holland's view of complex adaptive systems. Instead of jumping from agents to emergent behaviors directly, this model first categorizes *species* of agents based on morphological or physiological function.

For example, if one had an ant simulation then the queen ant would reside in a *species* by herself, then worker ants engaged in foraging would be in another, and ants engaged in brood sorting in another, and so on. These groups, working concurrently towards different goals, then combine into an aggregated emergent behavior, e.g. shortest-path building, construction of new spires in termite colonies, etc.

The world view, shown in Figure 41, creates an interconnected hierarchy. Starting at the bottom level, agents process their current system state, this includes both the actual environment as well as signals from other agents, and decide what function to perform. Internal bias guides the agent towards a specific function. Communities of agents, i.e. *niches*, will form, change, and dissolve, as agents move from one community to the next. These changes manifest as changes in emergent behaviors as one moves up the hierarchy.

Feedback then flows down the hierarchy at every time step, leading to behavior transitions, completing a cycle of system feedback.

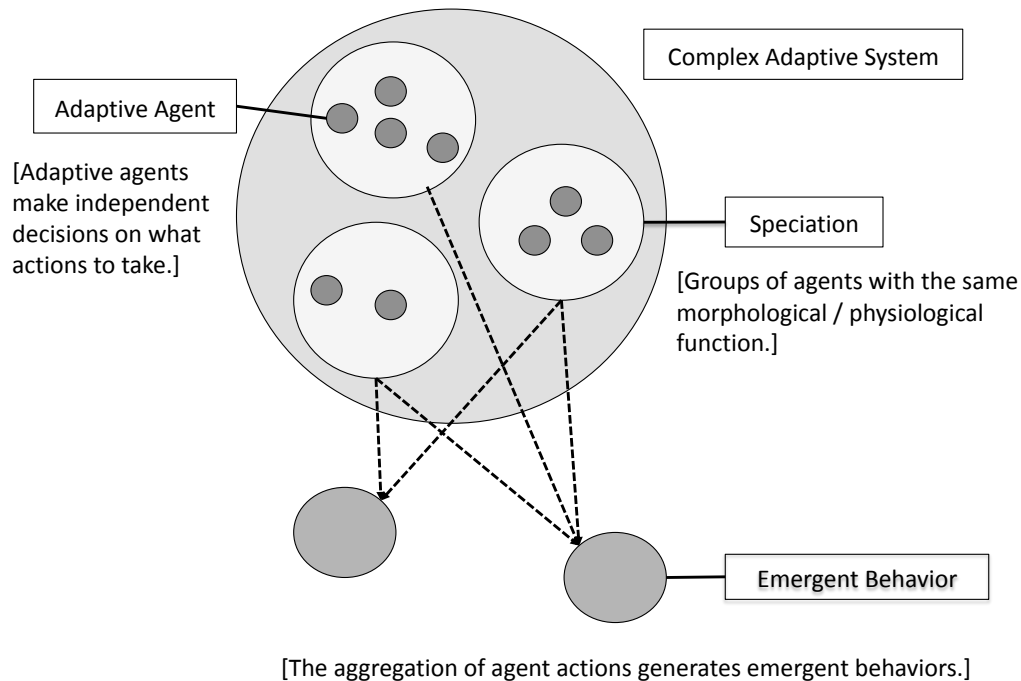


Figure 40. Updated view of a CAS.

The importance of speciation at the bottom level cannot be overstated. It allows agents to immediately adapt to their environment. If an ant decides more workers are needed to retrieve food, then it will opt to retrieve food. Otherwise, it may decide to clean up the nest. The ant's decision is based on its current understanding of the local system state, coupled to internal bias. Its decisions functionally align it to a group dedicated to a goal, e.g. food retrieval, refuse cleanup, etc. These groups aggregate into macro-level, emergent behaviors where shifts in membership create shifts in macro-level behavior patterns. The proposed world-view model takes into account both environmental and agent induced changes and their effect on emergent behaviors.

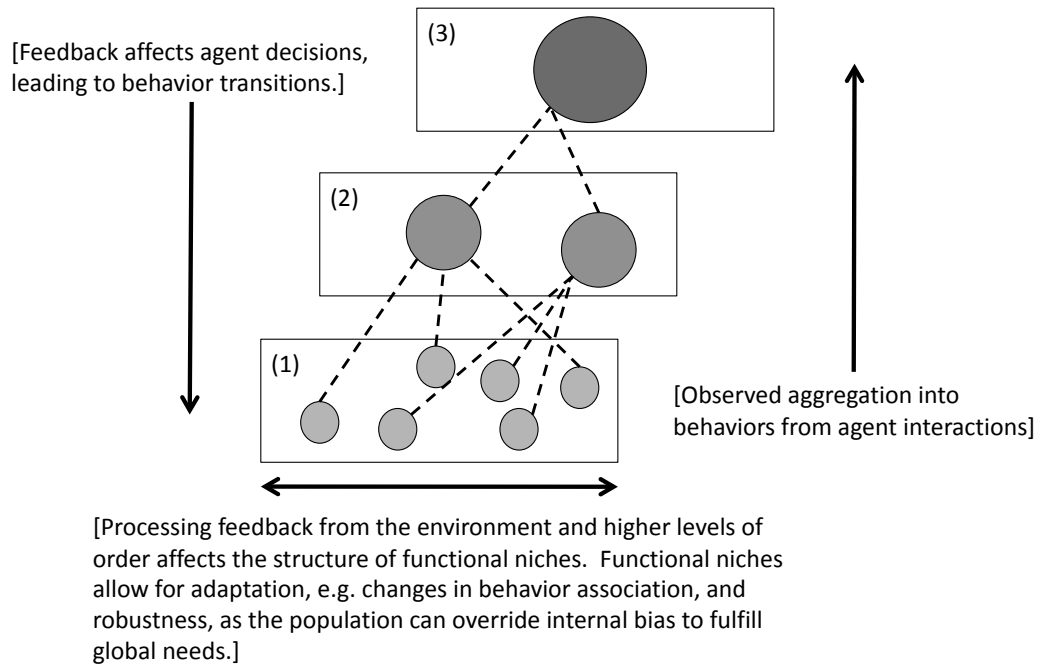


Figure 41. Updated world view. Agents at the micro-level align (1) to functional communities (2) which can combine to form macro-level emergent behaviors (3). Feedback flows up and down the hierarchy, creating changes in functional niches, which then creates macro-level change, in an eternal cycle of causal feedback loops.

8.6 Speciation Algorithm (SEA)

This dissertation's focus is on providing models and algorithms that can be applied in engineered systems that allow one to design systems with desired emergent behaviors, e.g. division of labor, without the need for a centralized controller or necessitating heavy heuristic guidance formulated by subject matter experts. In other words, design systems that are highly adaptable and robust *online*, with minimal outside guidance, based upon theorized properties of emergent systems and agent speciation.

To meet this need, this section introduces the Speciation Algorithm (SEA). The SEA serves as method for designing a system to create desired emergent behaviors. As these

behaviors are the manifestation of macro-level goals, then one can see the SEA as a top-down, design approach similar to Breddin [15]. However, agent design in the SEA couples micro-level actions, i.e. agent niches aligned to subgoals, to macro-level behaviors, i.e. system goals. Thusly accounting for both macro-micro and micro-macro feedback loops.

The SEA is based upon the eDEVS framework developed by King and Hodson [75]. The main insight from King and Hodson's framework is that multi-agent systems work towards multiple goals simultaneously. When moving up the hierarchy, niches of agents dedicated to specific micro-level goals aggregate into complex, macro-level behaviors. Although King and Hodson [75] used a formal framework to couple these niches together, the approach is generalizable.

The SEA (Algorithm 3) begins at the top level of the system requiring the engineer to develop all system level goals agents should accomplish, i.e. determine the macro-level behaviors the engineer wants the system to express. Each goal is then sub-divided into other goals, and this continues until the engineer is satisfied that all possible sub-goals are accounted for and tied to higher-level ones.

Algorithm 3 : Speciation Entropy Algorithm

Input: System Requirements

Output: None

```

1: for all g in Goals do
2:   for all sub_g in g do
3:     design_agent(g)
4:   end for
5: end for
6: Execute System

```

Next, agents are designed with the necessary capabilities to achieve sub-goals. These sub-goals can be as simple as pick up a resource and drop it at a specific location. Agents can be morphologically different, i.e. some designed to carry out specific tasks, or physiologically different, i.e. internal bias thresholds or parameters ensures variety in population decisions. Furthermore, agents must possess an internal decision function. The design

of which is left up to the engineer, however, these functions necessarily need to be easily calculable for real-world dynamic changes could make solutions to long-term decision methods obsolete before execution.

Any SEA implementation will necessarily define goals and then recursively break those goals into related sub-goals. Once one reaches an atomic sub-goal, i.e. a goal that cannot be broken down any further, then one begins agent design.

Agents will need to be capable of accomplishing the smallest sub-goals in the system. This enables them to functionally align at runtime, creating the first hierarchical level of niches in the system. As agents, and by extension their niches, can freely interact, one must carefully define the types of signals and receptors agents possess. This restricts signals between agents and niches, allowing one to trace causal chains to emergent, macro-level behaviors.

Building a house serves as an illustrative example.³ The desired macro-level outcome is a house which one breaks into sets of micro-level sub-goals, e.g., plumbing, electricity, framing, roofing, etc. These micro-level sub-goals possess associated tasks. Single agents, or small teams of agents, possess the skills necessary to accomplish simple tasks, such as nailing a piece of wood, pouring concrete, etc., that aggregate to the accomplishment of a micro-level goal. Over time, agents complete more and more micro-level sub-goals, eventually meeting the desired macro-level outcome.

Allen's [3] work on Hierarchy Theory posits that lower levels of a hierarchical system present dynamic possibilities, scoping what a system can accomplish, while higher levels provide a purpose that lower levels work towards. The SEA follows this theoretical idea. Agents, the smallest component of a multi-agent system, define the possible functional niches a system may contain. Niche populations may be dynamic, but only niches with associated sub-goals will ever occur during runtime. However, these sub-goals are defined by

³The following example can be found in [75]

macro-level, system goals. In other words, macro-level goals provide *purpose* to lower levels. The ebb and flow of system resources and agent decisions creates a dynamic feedback loop that moves a system towards desired goals.

8.7 Domains of Interest

Three main domains were implemented to showcase the SEA. The first domain, base defense, consists of multiple defenders trying to prevent an invader from reaching their home. Modeled as a Pursuit-Evasion Game (PEG), the defending agents use a role-based policy to instantiate different *species* in the population. In this manner, the agents can collectively defend their space from a superior invader.

In the second domain, k coverage, both morphological and entropy-based guided agents are tested. The k coverage domain explores how entropy-based agents create self-organized *niches* in response to environmental stimuli. Additionally, results from experiments in the domain show how morphological differences can have a positive impact on system performance.

Finally, this work extends previous explorations of division of labor in multi-agent systems. In this final section, experiments explore how speciation creates robust systems. Static, optimized populations serve as a basis of comparison for robustness measures. Results empirically support the hypothesis that speciation can aid in building hierarchical, self-organizing, robust systems capable of meeting multiple macro-level goals free from a centralized controller.

Pursuit-Evasion Games.

Pursuit-evasion games (PEGs) represent a class of optimization problems where a team of pursuit agents tries to capture a team of evading agents [23, 2]. PEGs contain four main components: 1) the environment in which pursuers and evaders play the game; 2) infor-

mation available to either team; 3) control laws governing how the agents move; and 4) a formal definition of what a *capture* means [23]. Although relatively easy to describe, PEGs contain many interesting research problems. For example, as PEGs contain teams with competing goals one can test how cooperation, or non-cooperation between agents affects the outcome. One could also test the advantages of agent signaling and even deception if evaders are allowed to send false signals into the environment. These types of behaviors can naturally lead to learning agents and collaboration.

The PEG presented by King, Bindewald and Peterson [73] had one evading agent and a team of pursuit agents. The evading agent's goal was to reach a central point in a two-dimensional (2D) grid before any of the pursuit agents could collide with it. The collision counted as a pursuit *capture* and resulted in a win for the pursuit team. To even out the odds a bit, evading agents possessed superior speed and small maneuver capabilities to avoid capture. This change also made the PEG particularly unique as most PEGs establish faster pursuit agents to guarantee a *capture* in a limited amount of time. Since the starting location of the evading agent was set randomly, pursuit agents needed to patrol the area around their nest site until they detected the evading agent, thus pursuit agents needed to accomplish two tasks concurrently: patrol and pursuit.

In order to meet this need, King, Bindewald and Peterson [73] introduced the Informal Team Assignment Algorithm (ITAA). The ITAA worked on the premise that agents choose roles dependent on their local, current system state. For the simple PEG described above, the pursuit agents held only two roles: patrol or pursuit. The threshold for pursuit was set to 1, as *capturing* an enemy agent is how pursuit agents won the game. Once an evading agent was detected, the agent assumed the pursuit role, and if enabled, signaled the location of the evading agent to agents nearby. If the evading agent left sensor range, then the pursuit agents would revert back to patrol.⁴

⁴One can find the entire experimental write up in [73].

Conceptually, King et al. [73] implemented ITAA by using the SEA. They first identified the primary goal of the system: capture invading agents. The main goal consists of two sub-goals: detect and intercept. As these were the smallest sub-goals, King et al. [73] designed agents with the ability to accomplish them. The agents in these experiments were morphologically the same, but were able to physiologically transition to specific roles depending on the current system state. Here, agents could only construct two types of functional niches: patrol and pursuit. The number of agents in either niche depended upon evader detection and any permitted signaling capabilities. Inter-agent communication created interesting, emergent behaviors.

First, agent role switching and signaling led to the formation of small teams pursuing agents that invoked temporary leadership behavior noted in pigeon flocks [21] where the agent that detected the evader first, created a small team of followers, all heading towards the evading agent. This is notable as it occurred without pre-defined leader-follow roles. Additionally, pursuit agents formed tactical nets around evading agents through self-organized flanking maneuvers (see Figure 42). Again, an emergent behavior not pre-programmed into the system. Finally, the approach enabled division of labor as agents completed two tasks, patrol and pursuit, simultaneously. In this domain, the various roles agents could switch between created adaptive speciation in the simulations. The ability of agents to quickly switch between roles enabled self-organized team formation.

Although a very simple system, all the components of an emergent system appeared during simulation. One, micro-level physiological switches created macro-level emergent behaviors and hierarchies, e.g. leader-follower interactions and flanking, in real-time and; two, the system showed adaptability as agents moved between functional niches as the environment changed. Importantly, agents remaining independent, choosing their functional role based on their local-system state, proves the scalability of the decentralized ITAA solution. However, the entire algorithm relies on the notion of agent speciation to work.

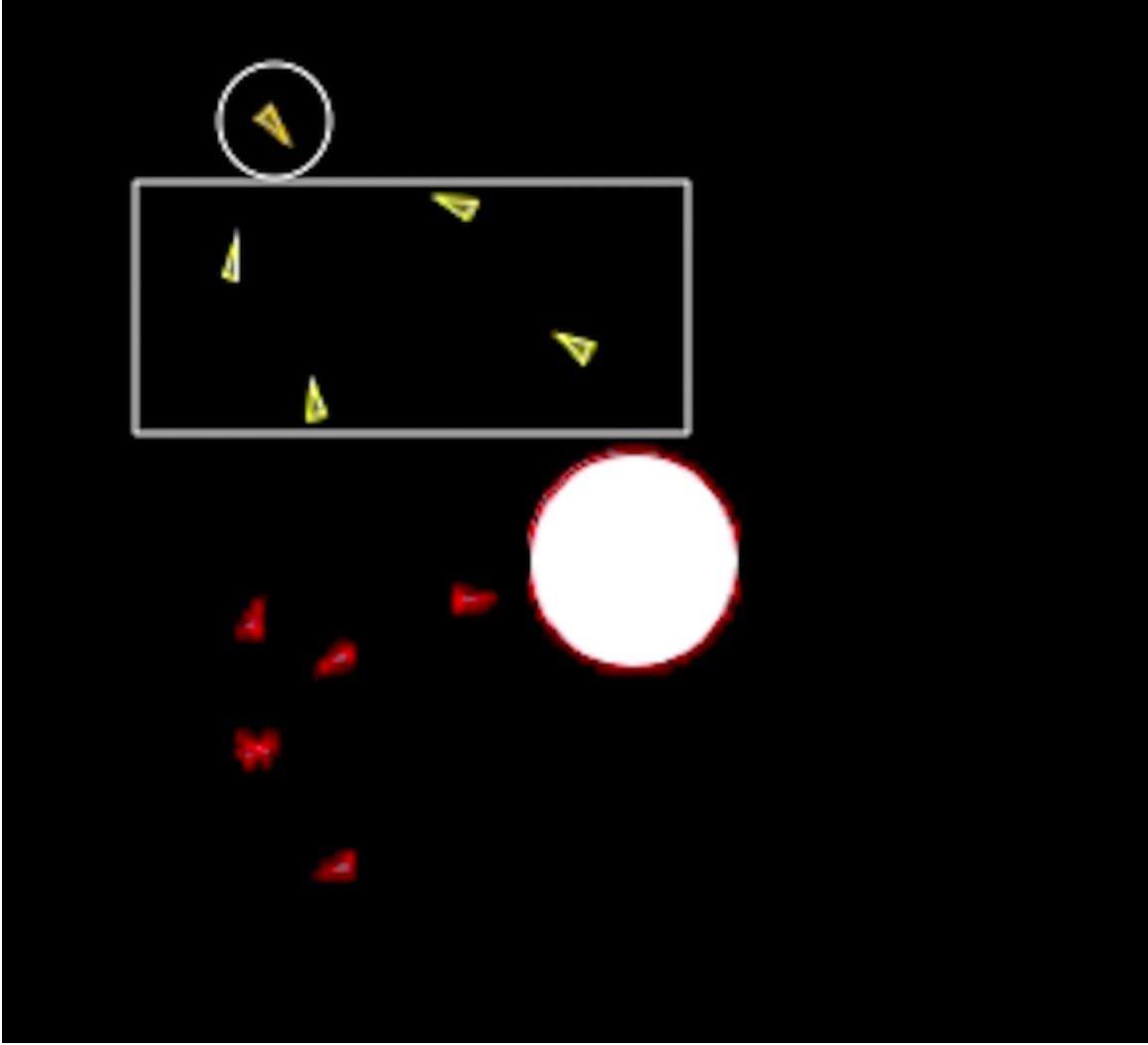


Figure 42. During this simulation, four pursuing agents in *pursuit* roles (boxed) converge on an evading agent (circled). Other pursuing agents remain in their current roles (*circle* or *patrol*). The white dot represents the nest. [73]

K Coverage Problem.

The k coverage problem centers on a population of agents whose task is to discover random targets in a set area and assign k agents to each discovered target, where k is a pre-defined number of agents. The problem combines the Cooperative Multi-robot Observation of Multiple Moving Targets (CMOMMT) introduced by Parker and Emmons [104] and the k -coverage problem associated with sensor networks proposed by Huang and Tseng [59]. Similar to PEGs, agents usually work in a cooperative manner to ensure the system, i.e.

the swarm of robots, is able to: 1) find targets, and 2) assign the appropriate k -number of agents to them. There is an interplay between target discovery and coverage that the robot team must balance. Over-assigning agents to a target effectively reduces both the number of agents available to cover newly discovered targets and reduces the sensor range of the entire swarm as agents stay with their targets until the targets move out of the assigned area.

A few researchers, in a quest to find a balance between search and coverage, have proposed different control methods. For example, Werger and Matarić [135] assigned weights to known targets based on the number of robots and targets in the area. Swarm agents would signal their availability to engage the target, allowing for team coordination. However, all targets and agent assignments were known *a priori* making for an unrealistic representation of real-world scenarios. Jung and Sukhatme [67] proposed robots calculating target weights at run-time to allow robots, sharing information with the collective, to self-distribute through the space. Kolling and Carpin [78] allowed agents to request help to create longer loiter times for agents over assigned targets as well as adding a signaling feature to ask for assistance when new targets appeared. These later approaches allowed run-time adaptation but did not investigate the effects of extending sensor ranges by adding a morphologically different agent to the mix.

King, Esterle and Peterson [74] extended previous work and tested both homogeneous and heterogeneous swarms engaged in a two-dimensional, k -coverage domain. Heterogeneous teams consisted of *observer* and *tracker* agents. Observer agents could not follow targets, however, they possessed a larger sensor range and could pass coordinates of found targets to any tracker agents nearby. Tracker agents could both look for targets in the search space and self-assign to cover any discovered targets. To enable proper k -coverage demands, tracker agents closer to known targets sent a suppression signal, a behavior suppression technique noted in bees [107], to keep excess trackers away. In this manner, the authors investigated the impact of heterogeneity, swarm size, and sensor ranges on goal

accomplishment.

Again, similar to King, Bindewald, and Peterson's [73]'s approach for PEGs, the components of SEA can be seen in King, Esterle, and Peterson's [74]'s methodology. First, they established macro-level goals and micro-level sub-goals before designing their agent populations. By identifying shortfalls in previous work, the authors modified agent capabilities, e.g. added suppression capabilities, as well as how to extend knowledge sharing between agents in the environment, i.e. creating observer agents. The authors were then able to compare the performance of physiologically and morphologically speciated populations.

Figures 43 and 44 show the average detection and average k-n saturation rates for homogenous and heterogeneous teams. Figure 43 clearly shows that most homogenous teams, i.e. those without observers, were able to find more targets except for a team of only five agents. This indicates that larger teams of just *tracker* agents do no benefit from having additional, specialized agents in the swarm. However, Figure 44 shows that *observer* agents did increase the likelihood of achieving the desired saturation rate across all know targets. There is an apparent trade-off between the homogeneous and heterogeneous teams.

However, both types of teams used an adaptive entropy gradient for agent decisions. Each agent updated local entropy scores associated with each target in the area, with each agent selecting the target with the highest entropy to cover. The entropy calculation ensured an even distribution of agents across targets that the signal suppression behavior repressed when targets became saturated, i.e. a target was already covered by k , closer agents. This allowed each agent to change its function decision online, creating dynamic teams of *search* and *cover* agents. So although the introduction of morphological differences did not yield a positive effect on detection rates, adaptive switching between these two roles still enabled swarm success as both types of teams consistently found and tracked targets in their associated areas. Again, highlighting the ability of highly diverse swarms to rapidly transition between behaviors without centralized control.

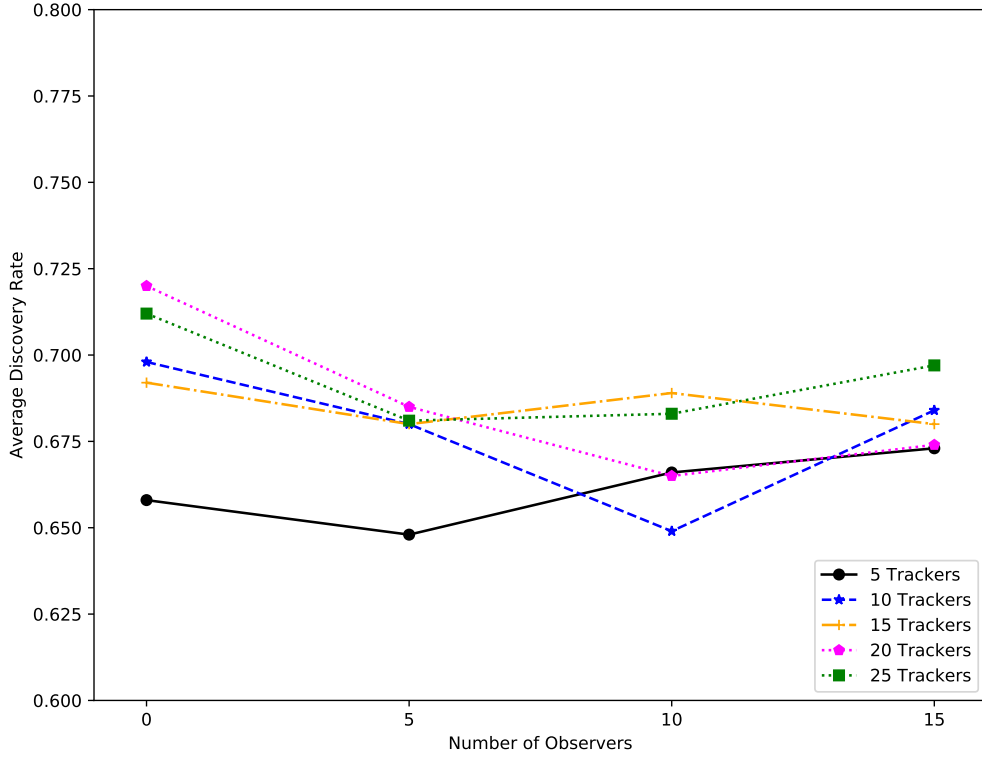


Figure 43. Average detection rates across all sensor ranges with teams of n trackers and m observers (x-axis), $k = 3$. [74]

Upon reviewing the data, the authors propose that the greater *observer* signal ranges created an attractive force for multiple *tracker* agents to flock to a region around an observer. These pockets of observer and tracker agents effectively reduced the sensor range of the entire swarm but increased the likelihood of achieving desired saturation rates. This work highlights some of the potential limits of morphologically-based systems, i.e. systems with highly specialized agents. The addition of a physically-specialized agent comes at a cost, in this case, it created lower detection rates. It also limits the possible niche transitions. Observer agents always reside in the *observer* niche, without the capability of moving to the *tracker* function, thereby reducing the plasticity of the overall system. However, adding specialized agents did enable the system to come closer to achieving its overall goal of as-

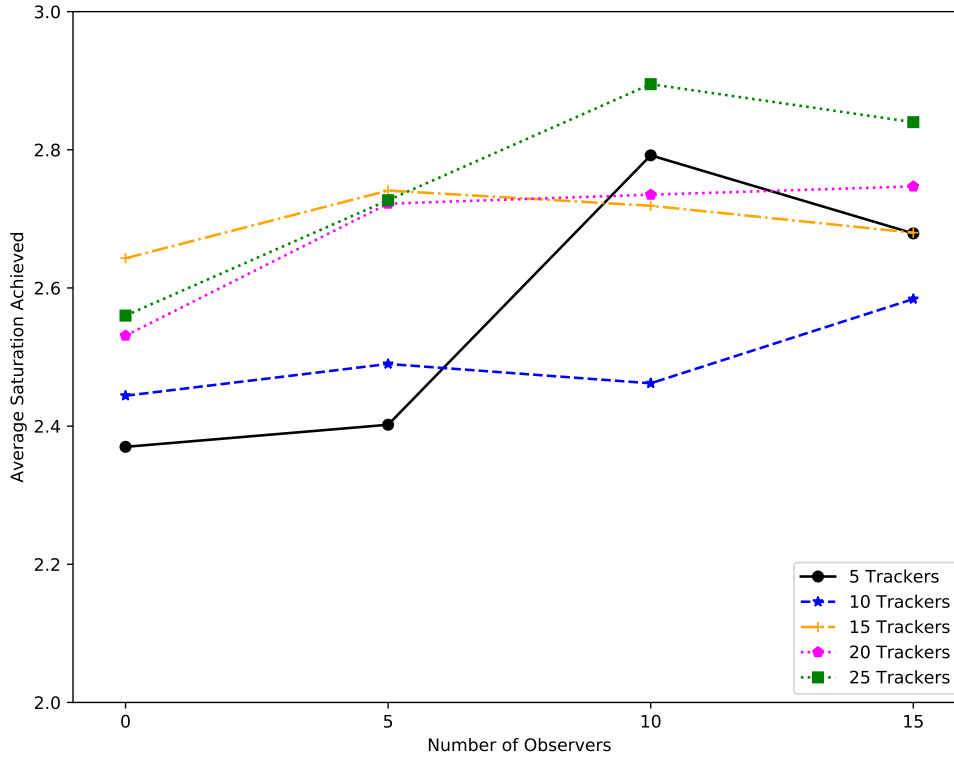


Figure 44. Average saturation rates across all sensor ranges with teams of n trackers and m observers (x-axis), $k = 3$. [74]

signing agents to targets at the desired 3:1 ratio. This means that physically-specialized agents can enhance system performance. Engineers then must be careful to balance the potential benefits of adding these agents with the apparent cost in both system robustness and adaptability.

8.8 Division of Labor

The previous section presented SEA approaches that created hierarchical, self-organized systems with multi-task capabilities free from a centralized controller. Although supporting two of the three major portions of this work's hypothesis, these examples failed to compare the robustness of speciated and non-speciated populations in a dynamic environment. This

section extends Duarte’s [33] and King and Peterson’s [77] work on the emergence of division of labor in multi-agent systems. Experiments in [33] and [77] showed that division of labor could emerge in a population of agents by evolving internal agent threshold values using a genetic algorithm. These internal agent values create a speciated population that is able to optimize to the task at hand. However, these works never compared resource output to non-speciated populations where all individuals in the system held the same internal threshold values. This section presents a series of experiments set in Adam Smith’s Pin factory to compare the robustness of speciated and non-speciated populations undergoing dynamic, environmental change.

Pin Factory.

In the simulated Pin Factory, a population of software agents, $A = \{a_1, a_2, \dots, a_n\}$, is tasked with producing as many pins as possible for a set time duration. An agent, a_n , can be defined by the tuple, $(f(x), t_n)$, where $f(x)$ represents its decision function and t_n represents its current task. At each time step, an agent must select a task from a set of nine, distinct tasks, $T = \{t_1, t_2, \dots, t_9\}$. As agents are homogeneous, factory agents possess the same internal decision function, either FTM (17) or LTM (18), and can accomplish every task. The set of tasks, T , require resources to complete. When a task, t_n , is completed, resources for task, t_{n+1} , immediately become available. Only when task, t_9 , is completed, does the population produce a ‘pin.’ Unless otherwise noted, a set of ten resources for task, t_1 , was added at every time step to ensure workers had at least one task available.

Environment.

Environment variables, *rich* and *constrained*, refer to task availability. All tasks were available to agents at every time-step in *rich* environments, while in *constrained* environments, tasks required resources. As noted in [33], task switching costs (i.e. time to com-

plete tasks) can influence the emergence of division of labor. In time-to-task length one (TT1), environments, every task took one time-step (ts) to complete. In time-to-task length variable (TTV) environments, tasks had different time-step requirements and were assigned by the following rule: $t_1 = 1, t_2 = 1, t_3 = 1, t_4 = 2, t_5 = 3, t_6 = 3, t_7 = 4, t_8 = 2, t_9 = 1$, where $t_1 = 1$ is interpreted as task one takes one time-step to complete, $t_4 = 2$, requires two time-steps to complete, and so on.

Simulations.

Each simulation followed the same process. After loading the initial 50 colonies, a colony processed as many pins as possible in a 360 time-step epoch. Agents selected a task to accomplish in a random order at each time-step. Once complete, the simulation ran the next colony through the same process. Each colony received a fitness score upon completing its epoch. The fitness function divided the number of pins produced by 48,000 - the number of pins Smith stated could be produced by 10 workers - creating a percentage, ρ . The simulation then added ρ percentage of workers from the colony to a population pool.

Workers selected for the pool underwent both crossover and mutation procedures outlined in [33]. Note: workers were pulled randomly from the population, meaning individuals were not individually selected for fitness, rather, it provided an indirect fitness function on colony production; avoiding individual agent optimization. Finally, the simulation created 50 new colonies (i.e. the next generation) by randomly selecting N workers from the pool, with N equaling the number of workers the colony should contain.

At each time-step, t , agents used one of two assigned decision functions (i.e. models): Bonabeau's fixed threshold model (FTM), discussed earlier, or the derived entropy based linear threshold model (LTM). Bonabeau's FTM [13] (Equation: 23) is widely used in biological modeling.

$$T_{\theta}(S) = \frac{s_i^n}{s_i^n + \theta_i^n} \quad (23)$$

In the FTM, the probability of performing a task is based on the internal agent task threshold, θ , the related task stimulus, s , and a steepness threshold, n . Accordingly, if $s \ll \theta$, the probability of performing the task is almost 0, while if $s \gg \theta$, then it is close to one [13]. In each experiment, task stimuli, s_i , changed at every time step according to $s(t+1) = s(t) + \delta_i - \alpha_i * A(t)$, where a task's increase in stimulus, δ_i , when not selected is offset by a decrease in stimuli, α_i , upon selection. $A(t)$ equals the proportion of agents currently working task t . Internal agent stimuli, θ_i , is randomly selected from the interval $(0, 1]$ and remained constant through the colony's lifecycle. The parameters α_i , δ_i and n were set to 3, 1, and 2 respectively.

The LTM was based on previous work with entropy-based agent decision processes [105, 141, 90, 76] and is the second decision function (Equation 24). The LTM uses task entropy, H_i , task stimulus, s_i , and the agent's internal task stimulus, θ_i , to produce a sum for a task. The agent selects the task with the highest sum.

$$f(s) = H_i + s_i + \theta_i \quad (24)$$

Task entropy, H_i , was calculated using Shannon's entropy (Equation 25) derived in [118]. The update rule for stimuli was changed to $s(t+1) = s(t) + 0.5$ if not selected, and $s(t+1) = s(t) - 0.5$ if selected. Stimuli values were capped between $[0,1]$ as they could quickly overwhelm agent bias, θ_i .

$$H_i = - \sum_i p_i \log_2 p_i \quad (25)$$

A task's entropy, H_i , was calculated by adding the current agent to the number of agents currently assigned to task i and dividing by all agents in the simulation, yielding p_i . Tasks with lower numbers of assigned agents have entropy values close to 1, while heavily saturated tasks have values close to 0. Task entropy provided a tie-breaking mechanism for any agents who had similar task stimuli and bias values for multiple tasks. Although a task could have a low entropy score, agents could still select it if task stimulus and/or task bias values were high enough.

Experiment.

The following experiments tested the impact of speciation on system robustness when the system undergoes a dramatic, environmental shift. Three different populations were initialized during set-up. The first two populations consisted of 50 colonies a piece. Each population differed only in their agent decision function where one set used Bonabeau's FTM [13] and the other set used King and Peterson's LTM [77]. The internal stimuli values, θ_i , were randomly initialized for each agent in each colony, creating speciated colonies. The third population used Bonabeau's FTM and was set up similarly, however, agents within a colony were clones of one another, meaning their internal stimuli values were exactly the same, creating an un-speciated set of colonies.

As described in Section 8.8: Simulations, colonies completed a 360 time-step epoch for two-thousand generations. During the initial one-thousand generations, all tasks took one time step to complete, and resources for each task were always available. At one-thousand generations, all resources were removed and tasks took a variable amount of time to complete, creating a dramatic shift in the environment. In this new environment, agents needed to complete a task, t_n , to create resources for the next task, t_{n+1} . Resources for task, t_1 , were always available, otherwise, the agents could not produce any pins.

During the first set of experiments, agents from the best performing colonies were selected to populate the next generation, allowing colonies to adapt and optimize to the environment. In the second set of experiments, the best performing colonies from the first set of experiments were selected to populate the next set of 50 colonies. Unlike previous experiments, these colonies remained constant throughout the duration, meaning they could not optimize to the environment. Setting up the experiments in this manner allowed for a direct comparison of evolutionary, non-evolutionary, speciated, and un-speciated populations.

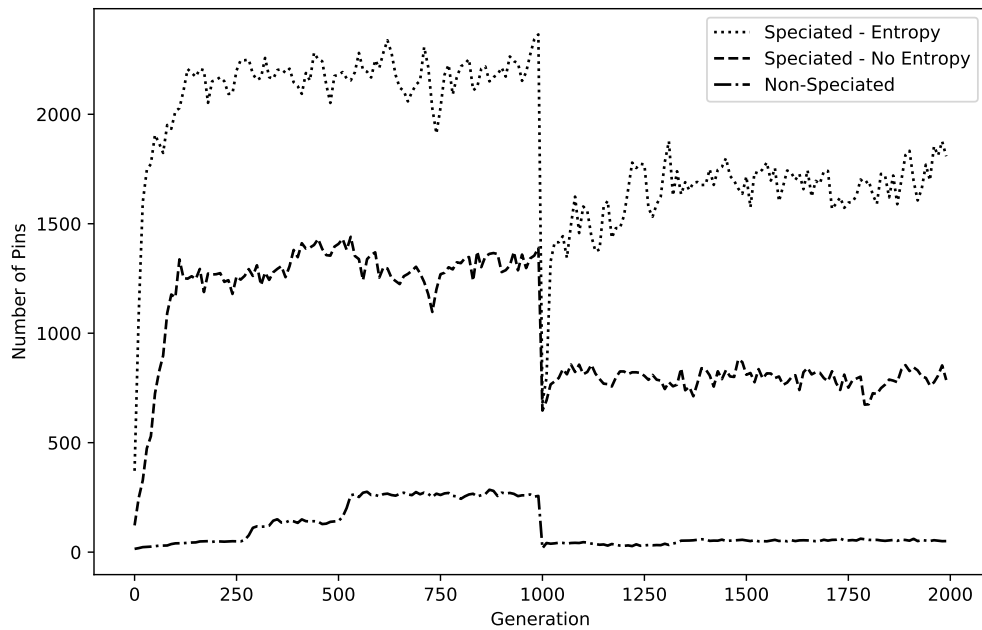


Figure 45. Average pins produced by three, evolutionary populations. Entropy populations used the LTM, while non-entropy populations used the FTM.

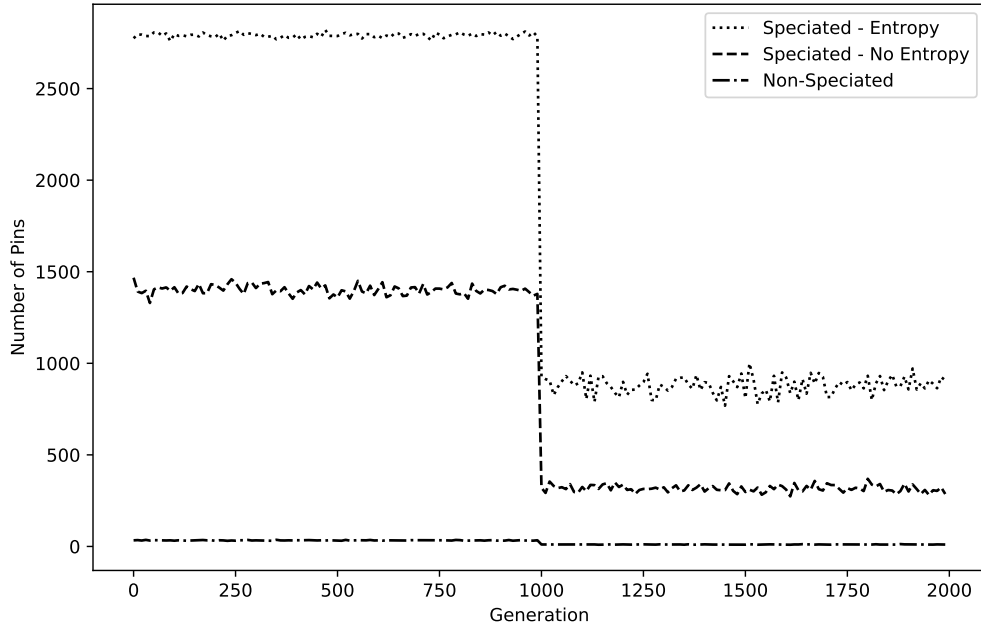


Figure 46. Average pins produced by three, non-evolutionary populations. Entropy populations used the LTM, while non-entropy populations used the FTM.

Figure 45 shows data from the first set of experiments with evolutionary populations. Results clearly show that speciated populations outperform non-speciated ones across all time-steps. Furthermore, both speciated populations optimize faster to their environments, and can be attributed to having more species available to optimize with, i.e. a more diverse pool of potential candidates, than the non-speciated population. It is also clear that entropy-based LTM colonies were more robust than the FTM ones, showing a slightly positive increase in pin production for the last one-thousand time steps while FTM colonies production remained flat. Un-speciated populations showed a similar trend but fell way short of the pin production of speciated populations.

Figure 46 present the results from the second set of experiments. Each speciated population outproduced their evolutionary counterparts in the first one-thousand time steps. This indicates that optimizing populations in a more stringent environment can lead to better performance in less exacting conditions. However, after the environmental shift at

one-thousand time steps, neither population produces the same average number of pins they did previously in the same environment. These results suggest that an evolutionary mechanism for agent speciation leads to better system outcomes. The non-speciated populations do not outperform their evolutionary version at any time step, showing that speciation does lead to more robust and adaptable systems.

8.9 Conclusion

This chapter presented the speciation algorithm (SEA) and provided empirical support for this dissertation's hypothesis that speciation creates hierarchical self-organized systems, capable of meeting multiple tasks, that are more robust to environmental change. Results showed that speciation leads to a horizontal shifting of agent niches, i.e. groups of functionally similar agents. These micro-level shifts modify macro-level behaviors which, in turn, influence micro-level decisions, in a perpetual, feedback cycle, leading to a robust system. As macro-level behaviors rely on micro-level interactions, this work highlights the power of decentralized execution of agent decision making with respect to both morphological and physiological constraints.

This work was able to show the impact of speciation in multiple, simulated domains, grounding the application of emergence-based research, free from the shackles of philosophical debate [52]. Agents guided by speciation - morphological or physiological - self-organized into functionally aligned niches creating macro-level, emergent patterns. The addition of niche movement based on speciation, i.e. the population shifting between functional roles, adds an overlooked nuance to emergence research. Instead of the traditional, vertical feedback loop cycle found in models of emergent, hierarchical systems, this work showed how a top-down design approach that takes into account the connection between agents and system goals, at every level, creates a robust, emergent system.

IX. Conclusion and Future Work

By analyzing experimental results from multiple domains, this dissertation has shown that the combination of speciation with the beneficial characteristics of emergent systems, results in a generalizable, decentralized-control methodology for complex, multi-agent systems. It was shown that speciation creates system hierarchies composed of goal-aligned *niches*. As niche actions aggregate into more complex behaviors, more levels emerge within the system hierarchy, eventually resulting in a system that can meet multiple tasks and is robust to environmental changes. Speciation provides a powerful tool for creating goal-aligned, decentralized systems that are inherently robust and adaptable, meeting the scalability demands of current, multi-agent system design.

9.1 Summary of Findings

First, as many emergence researchers [13, 55, 57, 93, 96, 103, 108] have theorized, results support the premise that simple, micro-level interactions create complex, macro-level behaviors. In domains such as nest defense, pursuit-evasion games, and the k coverage problem, agents armed with simple sensors and decision making algorithms were able to accomplish system goals, e.g. intercept and capture an evader, find and follow targets of interest, etc., while exhibiting complex, *novel* behaviors. For example, agent flanking emerged in the nest defense domain although agents possessed only two roles. Similarly, in the PEG, agents guided by task stimuli and entropy calculations, self-organized into pursuit teams, achieving high rates of detection and capture.

Next, this work revisited informational entropy developed by Shannon [118] in 1948, extended by Schulman and Seiden [116] in the 1970s, and the subject of current research interest in guiding multi-agent systems [105, 94, 39, 141, 90]. Entropy served two purposes in this work, as a guide for agent decision making and as a basis for measuring system state.

In multiple domains, results showed that entropy-based decision making can lead agents to better solutions. It also created natural, functional divisions of labor in Adam Smith's Pin factory and the k coverage domain. Additionally, entropy formed the basis for a graphical representation of system state. These graphs showed how hierarchies develop and shift over time as the system state changes, creating a mechanism for monitoring system status over time and possibly a method for emergent behavior detection.

Overall, results indicate that micro-level decisions and interactions benefit from entropy-guided decision making methods. However, this could be due more to entropy calculations imitating agent learning and bias versus providing some *causal* force. In other words, the constant change of entropy scores with respect to the dynamic environment shortcut agent learning methodologies. Other approaches would have to be implemented and tested against baseline entropy-performance results to truly test entropy's potential benefit as a component in agent decision making in multi-agent systems.

Results also answer challenges laid out in previous work. One, speciation provides plasticity in a system and shows how agents can rapidly switch between different regimes of behavior without centralized control, a noted ability in collective systems [103]. Two, it is a building block for self-organized behaviors at the macro-level. This view is supported, although not explicitly, by researchers such as Corning, Holland and Wimsatt. Wimsatt states that emergence involves an ordering, or organization, of different parts [140]. Holland acknowledges the importance of diversity in his ecosystem view, and how it creates cycles of recirculation, intertwining various species into a mutual beneficial webs of resource harvesting and sharing [57]. One can also view speciation as a component of Corning's synergy [24], where as agents evolve their internal bias, they align in a meaningful way that creates emergent effects in a non-linear manner. Physiological speciation allows a horizontal movement of functionality across the micro-level, developing ordered niches which combine to form macro-level behaviors. The self-organization of agents into func-

tional niches, based on their morphological or physiological differences, creates Wimsatt's self-organized order and Corning's synergies coupled with Holland's diverse ecosystem model, playing a vital role in the emergence of system robustness and adaptability.

9.2 Recommendations

Future work can look at various ways to adjust the evolution of speciation in a population. For example, a maturing process where agents can learn to prefer certain tasks over others, influenced by environmental factors, can extend the work found in Chapter VI. This answers both a question posed by Duarte [33] and Darwinian [26] efforts at explaining how agent instincts and habits evolve. From a macro-level viewpoint, the evolution of desired tasks and traits will be influenced by both environmental factors and agent success. Successive generations will necessarily need to inherit successful traits but also be armed with the ability to change within their lifecycles. Some interesting comparisons could be drawn between physiological and morphological based populations and their performance in dynamic environments over time. The author theorizes that physiologically plastic populations would fare better in environments with strong adaptation demands while morphologically different populations may produce better system-level outcomes, e.g. resource production, agents captured, etc., in fairly stable environments.

One could also test the robustness of agents in Chapters III, V, and VII by adding agent aging or refueling to the simulations. In this manner, agents must adjust to the population shifting in number, having to cover tasks that suddenly become available. This can play into ideas of temporal polytheism [13] where agents shift preferred tasks, or gain knowledge to do other tasks, over time. In these simulations one can test the impact of aging on the population as well as test how an evolving heterogeneous group of different aged individuals impacts the emergence of different phenomena. Furthermore, implementing non-speciated populations of agents in these domains would provide a baseline for comparing system

performance and robustness.

Finally, the primary goal of emergence research is to create an all-encompassing framework that other researchers can use when engineering and studying emergent systems. Currently no overarching theory exists on how hierarchies in large, dynamic systems self-develop [30] nor is there a solid theory on how feedback between the levels creates emergent behaviors [57]. In addition to these two branches of study, philosophical debates on *strong* and *weak* emergent behaviors, and how relevant such debates are to the application of emergence research to many fields, still exists. Although this work chose to separate the two ideas, they are inextricably linked. Will researchers ever answer the true, underlying question: will all emergent phenomenon be explained? In other words, will the field find that all emergent behaviors are *weakly* emergent? Or will the field actually prove that *strong* emergent phenomena exist? If so, this holds serious consequences for our understanding of the universe and the models constructed to explain its behaviors [28]. As Chalmer's wrote, "Strong emergence, if it exists, can be used to reject the physicalist picture of the world as fundamentally incomplete." [28] Such a refutation would indeed overturn our understanding of the world.

9.3 Contributions to Knowledge

As the primary theme for this work, speciation served to expand the focus and study of emergence and its role in multi-agent systems. Over the course of this dissertation, results showed how this relatively simple concept held powerful implications for emergence research. Speciation, defined as morphological or physiological differences in an agent population, drives horizontal shifts in micro-level functions. These shifts lead to the appearance, modification, and disappearance of macro-level, emergent behaviors which, in turn, influence micro-level agent decisions. The key finding, that aligns with observations on natural systems, is that agent speciation creates agents with plasticity, which creates

highly robust and adaptable systems [13]. This work is the first to empirically explore this concept and its ties to emergence research.

Furthermore, this work presented the speciation algorithm (SEA) which provides a top-down design approach that ties speciation to theoretical emergence concepts. It provides an application of these concepts absent in other approaches. The SEA creates hierarchical, self-organized systems whose inherent emergent behaviors support the system's overall purpose. These systems are decentralized, can meet multiple task requirements and are robust to environmental change. This work bridges the gap between emergence theory and application.

Finally, by adding speciation as a property of emergent systems, this work adds to the current body of emergence research. It empirically shows that complex, emergent systems can be created with simple agents, doing simple things. Furthermore, it provides a basic component for organizational structure in an emergent system. It provides the first hierarchical level of organization, i.e. formation of functional niches, upon which macro-level behaviors emerge. Functional selection, i.e. speciation, forms a shifting landscape of micro and macro-level interactions and behaviors. The shifting landscape allows a system to be robust to variance in system composition and adaptable to dynamic shifts in the environment. It also establishes a fundamental component for instantiating decentralized approaches to controlling multi-agent systems which is scalable to the demands of current, multi-agent systems operating in dynamic environments.

Bibliography

1. “niche”. In *Merriam-Webster Dictionary*. Sep 2017.
2. Alexander Alexopoulos, Tobias Schmidt, and Essameddin Badreddin. Cooperative pursue in pursuit-evasion games with unmanned aerial vehicles. In *Proceedings of 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015.
3. Timothy F. Allen. A summary of the principles of hierarchy theory.
4. Gary An. Integrating physiology across scales and formalizing hypothesis exploration with agent-based modeling. *Journal of Applied Physiology*, 118(10):1191–1192, May 2015.
5. Nadia Aubin-Horth and Susan C. P. Renn. Genomic reaction norms: using integrative biology to understand molecular mechanisms of phenotypic plasticity. *Molecular Ecology*, 18(18):3763–3780, September 2009.
6. Fernando Barros. Dynamic structure discrete event system specification: A new formalism for dynamic structure modeling and simulation. In C. Alexopoulos, K. Kang, R. Lilegdon, and D. Goldsman, editors, *Proceedings of the 1995 Winter Simulation Conference*, pages 781–785, 1995.
7. Andrew Barto, Satinder Singh, and Nuttapon Chentanez. Intrinsically motivated learning of hierarchical collections of skills. In , pages 112–119, 2004.
8. Randal Beard, Timothy McLain, Michael Goodrich, and Erik Anderson. Coordinated target assignment and intercept for unmanned air vehicles. *IEEE Transactions on Robotics and Automation*, 18(6), 2002.
9. Mark Bedau. Downward causation and the autonomy of weak emergence. *Principia: an international journal of epistemology*, 6(1):5–50, 2002.
10. Mark A. Bedau and Paul Humphreys, editors. *Emergence: Contemporary Readings in Philosophy and Science*. MIT Press, 2008.
11. S. N. Beshers and J. H. Fewell. Models of division of labor in social insects. *Annual Review of Entomology*, 46(1):413–440, 2001.
12. Lachlan Birdsey, Claudia Szabo, and Katrina Falkner. Identifying self-organization and adaptability in complex adaptive systems. In *11th IEEE International Conference on Self-Adaptive and Self-Organizing Systems*, 2017.
13. Eric Bonabeau, Marco Dorigo, and Guy Theraulaz. *Swarm Intelligence: From Natural to Artificial Systems*. Oxford University Press, 1999.

14. Fabio Boschetti, Mikhail Prokopenko, Ian Macreadie, and Anne-Marie Grisogono. Defining and detecting emergence in complex networks. In *International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*, pages 573–580. Springer, 2005.
15. Iris Breddin. Self-organisation and emergence. Technical report, Technical University of Berlin, 2006.
16. Eamonn Butler. *The Condensed Wealth of Nations and the Incredibly Condensed Theory of Moral Sentiments*. ASI Research Ltd., 2011.
17. Lei Cao, He shun Tan, Hui Peng, and Ming cong Pan. Multiple uavs hierarchical dynamic task allocation based on pso-fsa and decentralized auction. In *2014 IEEE International Conference on Robotics and Biomimetics (ROBIO 2014)*, pages 2368–2373, 2014.
18. Manuel Castillo-Cagigal, Eduardo Matallanas, Iñaki Navarro, Estefanía Caamaño-Martín, Félix Monasterio-Huelin, and Álvaro Gutiérrez. Variable threshold algorithm for division of labo analyzed as a dynamical system. *IEEE Transactions on Cybernetics*, 44(12):2242–2252, 2014.
19. David J. Chalmers. Varieties of emergence, August 2002.
20. R. Chawla, P. Thakral, A. K. Kaura, and K. Gupta. Modified centralized approach for preventing collision at traffic intersection. In *2017 4th International Conference on Signal Processing, Computing and Control (ISPCC)*, pages 35–42, 2017.
21. Duxin Chen, Xiao-Lu Liu, and Hai-Tao Zhang. Switching hierarchical leadership in coordinated movement of pigeon flocks. In *Proceedings of the 35th Chinese Control Conference*, 2016.
22. P.C. Chu and J.E. Beasley. A genetic algorithm for the multidimensional knapsack problem. *Journal of Heuristics*, 4(1):63–86, 1998.
23. T. Chung, G. Hollinger, and V. Isler. Search and pursuit-evasion in mobile robotics. *Autonomous Robots*, 31(3):299–316, 2011.
24. Peter Corning. The re-emergence of “emergence”: A venerable concept in search of a theory. *Complexity*, 7.6:18–30, 2002.
25. Mario Couture. Complexity and chaos – state-of-the-art; overview of theoretical concepts. Technical Report 2006-453, DRDC Valcartier TM, 2007.
26. Charles Darwin. *On the Origin of Species by Means of Natural Selection or the Preservation of Favoured Races in the Struggle for Life*. Arcturus Publishing Limited, sixth edition, 2018.

27. Kerstin Dautenhahn. Embodiment and interaction in socially intelligent life-like agents. *Computation for Metaphors, Analogy and Agent, Spring Lecture Notes in Artificial Intelligence*, 1562:102–142, 1998.
28. P. Davies and P. Clayton, editors. *The Re-Emergence of Emergence: The Emergentist Hypothesis from Science to Religion*. Oxford University Press, 2006.
29. Viviane M. de Oliveira and Paulo R.A. Campos. The emergence of division of labor in a structured response threshold model. *Physica A*, 517:153–162, 2019.
30. Ada Diaconescu, Louisa Jane Di Felice, and Patricia Mellodge. Multi-scale feedbacks for large-scale coordination in self-* systems. In *2019 IEEE 13th International Conference on Self-Adaptive and Self-Organizing Systems (SASO)*, pages 137–142, 2019.
31. H. Duan, P. Li, and Y. Yu. 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, January 2015.
32. Junhua Duan, Yi an Zhu, and Bingzhe Li. Study and design of multi-robots pursuing based on improved ant colony labor division method. In *2012 IEEE International Conference on Computer Science and Automation Engineering (CSAE)*, pages 1–5, 2012.
33. Ana Leitao Fernandes Duarte. *Evolution of Self-organized Division of Labor in Social Insects*. PhD thesis, Rijksuniversiteit Groningen, 2012.
34. Mohamed Elhoseny, Alaa Tharwat, Xiaohui Yuan, and Aboul Ella Hassanien. Optimizing k-coverage of mobile wsns. *Expert Systems with Applications*, 92:142 – 153, 2018.
35. Lukas Esterle and Peter R. Lewis. Online multi-object k-coverage with mobile smart cameras. In *Proc. of the Int. Conf. on Distributed Smart Cameras*, pages 1–6. ACM, 2017.
36. Rick Everts, Frank Ritter, Paolo Busetta, and Jennifer Bittner. Cojack - achieving principled behavior variation in a moderated cognitive architecture. In *2008 Brims Conference - Behavior Representation in Modeling and Simulation.*, pages 80–88, 2008.
37. Rick Everts, John Thangarajah, Nitin Yadav, and Thanh Ly. A framework for modeling tactical decision-making in autonomous systems. *The Journal of Systems and Software*, January 2015.
38. Paul Fieguth. *An Introduction to Complex Systems*. Springer, 2017.

39. D. Fisch, M. Janicke, N. Sick, and C. Muller-Schloer. Quantitative emergence – a refined approach based on divergence measures. In *2010 Fourth IEEE International Conference on Self-Adaptive and Self-Organizing Systems*, pages 94–103, 2010.
40. G. Fusco and H. Gupta. Selection and orientation of directional sensors for coverage maximization. In *Proc. of the Int. Conf. on Sensor, Mesh and Ad Hoc Communications and Networks*, pages 1–9, 2009.
41. Jacques Gignoux, Guillaume Chérel, Ian D. Davies, Shayne R. Flint, and Eric Lateltin. Emergence and complex systems: The contribution of dynamic graph theory. *Ecological Complexity*, 31:34 – 49, 2017.
42. Rhys Goldstein and Gabriel Wainer. Devs-based design of spatial simulations of biological systems. In M.D. Rossetti, R.R. Hill, B. Johansson, A. Dunkin, and R.G. Ingalls, editors, *Proceedings of the 2009 Winter Simulation Conference*, pages 743–754, 2009.
43. Deborah M. Gordon. The organization of work in social insect colonies. *Complexity*, 8(1):43–46, 2003.
44. R. Gorelick, S. M. Bertram, P. R. Killeen, and J. H. Fewell. Normalized mutual entropy in biology: quantifying division of labor. *American Naturalist*, 164(5):678–682, 2004.
45. John Grefenstette, Rajeev Gopal, Brian Rosmaita, and Dirk Van Gucht. Genetic algorithms for the traveling salesman problem. In *Proceedings of the First International Conference on Genetic Algorithms and Their Applications*, pages 160–168, 1985.
46. Andreas Greven, Gerhard Keller, and Gerald Warnecke. *Entropy*. Princeton University Press, 2003.
47. Roderich Groß, Shervin Nouyan, Michael Bonani, Francesco Mondada, and Marco Dorigo. Division of labour in self-organised groups. In Minoru Asada, John C. T. Hallam, Jean-Arcady Meyer, and Jun Tani, editors, *From Animals to Animats 10*, pages 426–436. Springer, 2008.
48. X. h. Shi and Z. y. Hua. Study of multiple air vehicles and multiple-mission route planning in virtual battlefield. In *2008 Asia Simulation Conference - 7th International Conference on System Simulation and Scientific Computing*, pages 1232–1235, 2008.
49. Moeed Haghnevis. *An Agent-Based Optimization Framework for Engineered Complex Adaptive Systems with Application to Demand Response in Electricity Markets*. PhD thesis, Arizona State University, 2013.
50. F. He and Y. Yao. Maneuver decision-making on air-to-air combat via hybrid control. In *2010 IEEE Aerospace Conference*, pages 1–6, Big Sky, MT, 2010.

51. M. Hefeeda and M. Bagheri. Randomized k-coverage algorithms for dense sensor networks. In *Proc. of the Int. Conf. on Computer Communications*, pages 2376–2380, 2007.
52. Carl Hempel and Paul Oppenheim. *Aspects of Scientific Explanation and Other Essays in the Philosophy of Science*. Free Press, 1965.
53. C. Tate Holbrook, Colin M. Wright, and Jonathan N. Pruitt. Individual differences in personality and behavioral plasticity facilitate division of labour in social spider colonies. *Animal Behavior*, 97:177–183, 2014.
54. John H. Holland. *Hidden Order: How Adaptation Builds Complexity*. Addison-Wesley, 1995.
55. John H. Holland. *Emergence: From Chaos to Order*. Addison-Wesley, 1998.
56. John H. Holland. Studying complex adaptive systems. *Systems Science and Complexity*, 19(1):1–8, 2006.
57. John H. Holland. *Signals and Boundaries: Building Blocks for Complex Adaptive Systems*. MIT Press, 2012.
58. Karen L. Hollis. Ants and antlions: The impact of ecology, coevolution, and learning on an insect predator-prey relationship. *Behavioural Processes*, 139:4–11, 2017.
59. Chi-Fu Huang and Yu-Chee Tseng. The coverage problem in a wireless sensor network. *Mobile Networks and Applications*, 10(4):519–528, 2005.
60. Qiaoqiao Huang, Xiaoxia Li, Fangfang Huang, Ruilong Wang, Bqoqian Lu, Yide Shen, Zhiwei Fan, and Peiqun Lin. Nutrient addition increases the capacity for division of labor and the benefits of clonal integration in an invasive plant. *Science of the Total Environment*, 643:1232–1238, 2018.
61. Paul Humphreys. How properties emerge. *Philosophy of Science*, 64(1):1–17, 1997.
62. Segismundo Izquierdo and Luis Izquierdo. Strictly dominated strategies in the replicator-mutator dynamics. *Games*, 2(3):355–364, 2011.
63. Jennifer M. Jandt and Anna Dornhaus. Spatial organization and division of labor in the bumblebee *Bombus impatiens*. *Animal Behavior*, 77:641–651, 2009.
64. R. Jeanson, P. Kukuk, and J. H. Fewell. Emergence of division of labor in halictine bees: contributions of social interactions and behavioral variance. *Animal Behavior*, 70:1183–1193, 2005.
65. Raphaël Jeanson, Jennifer H. Fewell, Root Gorelick, and Susan M. Bertram. Emergence of increased division of labor as a function of group size. *Behavioral Ecology and Sociobiology*, 62:289–298, 2007.

66. J. S. Jennings, G. Whelan, and W. F. Evans. Cooperative search and rescue with a team of mobile robots. In *Proc. of the Int. Conf. on Advanced Robotics*, pages 193–200, 1997.
67. Boyoon Jung and Gaurav S Sukhatme. Cooperative multi-robot target tracking. In *Distributed Autonomous Robotic Systems 7*, pages 81–90. Springer, 2006.
68. Elsa Kania. Swarms at war: Chinese advances in swarm intelligence, July 2016.
69. Stuart A. Kauffman. *The Origins of Order: Self-Organization and Selection in Evolution*. Oxford University Press, 1993.
70. James Kennedy and Russell Eberhart. *Swarm Intelligence*. Academic Press, 2001.
71. Sami Khuri, Thomas Bäck, and Jörg Heitkötter. The zero/one multiple knapsack problem and genetic algorithms. In *SAC*, 1994.
72. Jaegwon Kim. Making sense of emergence. *Philosophical Studies: An International Journal for Philosophy in the Analytic Tradition*, 95(1/2):3–36, 1999.
73. David W. King, Jason M. Bindewald, and Gilbert L. Peterson. Informal team assignment in a pursuit-evasion game. In *The Florida Artificial Intelligence Research Society-31*, May 2018.
74. David W. King, Lukas Esterle, and Gilbert Peterson. Entropy-based team self-organization with signal suppression. In *The 2019 Conference on Artificial Life (ALIFE2019)*, 2019.
75. David W. King, Douglas D. Hodson, and Gilbert L. Peterson. A devs-based approach towards hierarchical ecosystem modeling. Submitted for publication consideration to *Simulation: Transactions of the Society for Modeling and Simulation International*, April 2019.
76. David W. King and Gilbert Peterson. A macro-level order metric for self-organizing adaptive systems. In *12th IEEE International Conference on Self-Adaptive and Self-Organizing Systems*, pages 60–69, 2018.
77. David W. King and Gilbert Peterson. The emergence of division of labor in multi-agent systems. In *13th IEEE International Conference on Self-Adaptive and Self-Organizing Systems*, pages 107–116, 2019.
78. Andreas Kolling and Stefano Carpin. Cooperative observation of multiple moving targets: an algorithm and its formalization. *The International Journal of Robotics Research*, 26(9):935–953, 2007.
79. Alexander Kossiakoff and William Sweet. *Systems Engineering Principles and Practice*. John Wiley and Sons, Inc., 2003.

80. Akhlesh Lakhtakia and Raúl J. Martín-Palma, editors. *Engineered Biomimicry*. Elsevier, 2013.
81. Christopher G. Langton. Studying artificial life with cellular automata. *Physica D: Nonlinear Phenomena*, 22(1-3):120–149, 1986.
82. Wonki Lee and DaeEun Kim. History-based response threshold model for division of labor in multi-agent systems. *Sensors (Basel)*, 17(6), Jun 2017.
83. Gavin M. Leighton, Daniel Charbonneau, and Anna Dornhaus. Task switching is associated with temporal delays in *temnothorax rugatulus* ants. *Behavioral Ecology*, 28(1):319–327, 2017.
84. Peter R. Lewis, Lukas Esterle, Arjun Chandra, Bernhard Rinner, Jim Torresen, and Xin Yao. Static, dynamic, and adaptive heterogeneity in distributed smart camera networks. *ACM Trans. Auton. Adapt. Syst.*, 10(2):8:1–8:30, June 2015.
85. Dongxu Li. Multi-player pursuit-evasion differential games, 2006.
86. Z. Li, D. Zhou, X. Zhao, H. Yang, L. Gui, and Y. Chang. Method for air combat formation optimization with uncertain information. In *2017 IEEE 2nd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, pages 1846–1854, 2017.
87. B. Liu, O. Dousse, P. Nain, and D. Towsley. Dynamic coverage of mobile sensor networks. *Trans. on Parallel and Distributed Systems*, 24(2):301–311, 2013.
88. De-Lin Luo, Chun-Lin Shen, Biao Wang, and Wen-Hai Wu. Air combat decision-making for cooperative multiple target attack using heuristic adaptive genetic algorithm. In *International Conference on Machine Learning and Cybernetics*, pages 473–478, 2005.
89. Lingzhi Luo, Nilanjan Chakraborty, and Katia Sycara. Provably-good distributed algorithm for constrained multi-robot task assignment for grouped tasks. *IEEE Transactions on Robotics*, 31(1):19–30, February 2015.
90. Richard P. Mann and Roman Garnett. The entropic basis of collective behavior. *Journal of the Royal Society Interface*, 12, 2017.
91. R. McCune, R. Purta, M. Dobski, A. Jaworski, G. Madey, A. Madey, Y. Wei, and M. B. Blake. 2013 winter simulations conference (wsc). In *Investigations of DDDAS for command and control of UAV swarms with agent-based modeling*, pages 1467–1478, 2013.
92. Marvin Minsky. *Society of Mind*. Simon and Schuster, Rockefeller Center, 1230 Avenue of the Americas, New York, NY 10020, 1985.

93. Melanie Mitchell. Complex systems: Network thinking. *Artificial Intelligence*, 170(18):1194–1212, December 2006.
94. M. Mnif and C. Müller-Schloer. Quantitative emergence. In *2006 IEEE Mountain Workshop on Adaptive and Learning Systems*, pages 78–84, 2006.
95. Sifat Momen and Amanda J.C. Sharkey. Strategies of division of labour for improving task efficiency in multi-robot systems. In *2009 World Congress on Nature & Biologically Inspired Computing (NaBIC)*, pages 672–677, 2009.
96. Mirko Morandini, Frederic Migeon, Marie-Pierre Gleizes, Christine Maurel, Loris Penserini, and Anna Perini. A goal-oriented approach for modelling self-organising mas. In *International Workshop on Engineering Societies in the Agents World*, 2009.
97. Felix Müller. Hierarchical approaches to ecosystem theory. *Ecological Modeling*, 63:215–242, 1992.
98. Wataru Nakahashi and Marcus W. Feldman. Evolution of division of labor: Emergence of different activities among group members. *Journal of Theoretical Biology*, 348:65–79, 2014.
99. Stefano Nolfi and Dario Floreano. *Evolutionary Robotics*. MIT Press, evolutionary robotics, evolution 2000.
100. I. M. Oliver, D. J. Smith, and J. R. C. Holland. A study of permutation crossover operators on the traveling salesman problem. In *Proceedings of the Second International Conference on Genetic Algorithms*, pages 224–230, 1987.
101. Santiago Ontanon, Gabriel Synnaeve, Alberto Uriarte, Florian Richoux, David Chruchill, and Mike Preuss. A survey of real-time strategy game ai research and competition in starcraft. *IEEE Transactions on Computational Intelligence and AI in Games*, 5(4), 2013.
102. Fatemeh Orooji, Hessam S. Sarjoughian, and Fattaneh Taghiyareh. Modeling and simulation of educational multi-agent systems in devs-suite. In *2010 5th International Symposium on Telecommunications (IST2010)*, pages 956–961, 2010.
103. Darren Pais. *Emergent Collective Behavior in Multi-Agent Systems: An Evolutionary Perspective*. PhD thesis, Princeton University, 2012.
104. L. E. Parker and B. A. Emmons. Cooperative multi-robot observation of multiple moving targets. In *Proc. of the Int. Conf. on Robotics and Automation*, volume 3, pages 2082–2089 vol.3, 1997.
105. H. Van Dyke Parunak and Sven Brueckner. Entropy and self-organization in multi-agent systems. In *Proceedings of the 2001 International Conference on Autonomous Agents*, pages 124–130, August 2001.

106. James Peters, Christopher Henry, and Sheela Ramanna. Reinforcement learning in swarms that learn. In *Proceedings of the International Conference on Intelligent Agent Technology*, 2005.
107. Anguluri Rajasekhar, Nandar Lynn, Swagatam Das, and P.N. Suganthan. Computing with the collective intelligence of honey bees – a survey. *Swarm and Evolutionary Computation*, 32:25 – 48, 2017.
108. Steen Rasmussen, Nils A. Baas, Bernd Mayer, and Martin Nilsson. Defense of the *ansatz* for dynamical hierarchies. *Artificial Life*, 7:367–373, 2001.
109. N. Razin, J.-P. Eckmann, and Ofer Feinerman. Desert ants achieve reliable recruitment across noisy interactions. *Journal of the Royal Society Interface*, 10(82), 2013.
110. C. W. Reynolds. Flocks, herds, and schools: A distributed model. *Computer Graphics*, 21(4):25–34, 1987.
111. Gene E. Robinson. Regulation of division of labor in insect societies. *Annual Review of Entomology*, 37(1):637–665, 1992.
112. Edmund M.A. Ronald, Moshe Sipper, and Mathieu S. Capcarrère. Design, observation, surprise! a test of emergence. *Artificial Life*, 5(3):225–239, 1999.
113. C. Rouff, A. Vanderbilt, W. Truskowski, J. Rash, and M. Hinchey. Verification of nasa emergent systems. In *Ninth IEEE International Conference on Engineering of Complex Computer Systems*, pages 231–238, 2004.
114. Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, third edition, 2010.
115. M. W. P. Savelsbergh and M. Sol. The general pickup and delivery problem. *Transportation Science*, 29:17–29, 1995.
116. L. S. Schulman and P. E. Seiden. Statistical mechanics of a dynamical system based on conway’s game of life. *Journal of Statistical Physics*, 19(3), 1978.
117. G. Serugendo, N. Foukia, S. Hassas, A. Karageorgos, S. Mostefaoui, O. Rana, M. Ulieru, P. Valckenaers, and C. Van Aart. Self-organisation: Paradigms and applications. In *Engineering Self-Organising Systems: Nature-Inspired Approaches to Software Engineering*, pages 1–19. Springer, 2004.
118. C.E. Shannon. A mathematical theory of communication. *The Bell Systems Technical Journal*, 27:379–423, 623–656, July, October 1948.
119. Yoav Shoham and Kevin Leyton-Brown. *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations*. Cambridge University Press, 2009.
120. Adam Smith. *Wealth of Nations*. William Strahan and Thomas Cadell, March 1776.

121. Jan-Phillip Steghöfer, Jörg Denzinger, Holger Kasinger, and Bernhard Bauer. Improving the efficiency of self-organizing emergent systems by an advisor. In *Seventh IEEE International Conference and Workshops on Engineering of Autonomic and Autonomous Systems*, pages 63–72, 2010.
122. Alexander Steiniger, Frank Krüger, and Adeline M. Uhrmacher. Modeling agents and their environment in multi-level-devs. In C. Laroque, J. Himmelspace, R. Pasupathy, O. Rose, and A.M. Uhrmacher, editors, *Proceedings of the 2012 Winter Simulation Conference*, 2012.
123. Ying Tan and Zhong yang Zheng. Research advance in swarm robotics. *Defence Technology*, 9(1):18–39, 2013.
124. Emmanuel Tannenbaum. When does division of labor lead to increases system output? *Journal of Theoretical Biology*, 247:413–425, 2007.
125. Danesh Tarapore, Dario Floreano, and Laurent Keller. Task dependent influence of genetic architecture and mating frequency on division of labour in social insect societies. *Behavioral Ecology and Sociobiology*, 64(4):675–684, February 2009.
126. Yentl Van Tendeloo and Hans Vangheluwe. Classic devs modelling and simulation. In W.K.V. Chan, A. D’Ambrogio, G. Zacharewicz, N. Mustafee, G. Wainer, and E. Page, editors, *Proceedings of the 2017 Winter Simulation Conference*, pages 644–658, 2017.
127. G Theraulaz, E Bonabeau, and J-L Deneubourg. Response threshold reinforcements and division of labour in insect societies. *Proc. of the Royal Society B: Biological Sciences*, 265(1393):327–332, 1998.
128. G. Theraulaz, J. Gautrais, S. Camazine, , and JL. Deneubourg. The formation of spatial patterns in social insects: From simple behaviours to complex structures. *Philosophical Trans. of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, 361(1807):263–282, 2003.
129. Sherif Tolba, Reda Ammar, and Sanguthevar Rajasekaran. Taking swarms to the field: Constrained spiral flocking for underwater search. In *2016 IEEE Symposium on Computers and Communications (ISCC)*, 2016.
130. James F. A. Traniello and Rebeca B. Rosengaus. Ecology, evolution and division of labor in social insects. *Animal Behavior*, 53:209–213, 1997.
131. Adeline M. Uhrmacher, Roland Ewald, Mathias John, Carsten Maus, Matthias Jeschke, and Susanne Bieermann. Combining micro and macro-modeling in devs for computational biology. In S.G. Henderson, B. Biller, M.-H. Hsieh, J. Shortle, J.D. Tew, and R.R. Barton, editors, *Proceedings of the 2007 Winter Simulation Conference*, pages 871–880, 2007.

132. Y. Ulrich, J. Saragosti, C. K. Tokita, C. E. Tarnita, and D. J. C. Kronauer. Fitness benefits and emergent division of labour at the onset of group living. *Nature*, 560(7720):635–638, 2018.
133. Rene Vial, Shahid Rashid, Cory Sharp, Omid Shakernia, Jin Kim, and Shankar Sastry. Pursuit-evasion games with unmanned ground and aerial vehicles. In *Proceedings of the 2001 IEEE International Conference on Robotics and Automation*, 2001.
134. Gabriel A. Wainer. *Discrete-Event Modeling and Simulation: A Practitioner’s Approach*. CRC Press, 2009.
135. Barry Brian Werger and Maja J. Matarić. From insect to internet: Situated control for networked robot teams. *Annals of Mathematics and Artificial Intelligence*, 31(1):173–197, 2001.
136. Danny Weyns. *Architecture-Based Design of Multi-Agent Systems*. Springer, 2010.
137. Mary. E. A. Whitehouse and Klaus Jaffe. Ant wars: combat strategies, territory and nest defence in the leaf-cutting ant *Atta laevigata*. *Animal Behavior*, 51(6):1207–1217, 1996.
138. E. O. Wilson. *The principles of caste evolution*. Sinauer, 1985.
139. Edward O. Wilson. The evolution of caste systems in social insects. *Proceedings of the American Philosophical Society*, 123(4):204–210, 1979.
140. William C. Wimsatt. Aggregativity: Reductive heuristics for finding emergence. *Philosophy of Science*, 64(4):372–384, 1997.
141. A.D. Wissner-Gross and C.E. Freer. Causal entropic forces. *Physical Review Letters*, 110(168702), April 2013.
142. Tom De Wolf, Tom Holvoet, and Giovanni Samaey. Engineering self-organising emergent systems with simulation-based scientific analysis. In *Fourth International Workshop on Engineering Self-Organising Applications*, pages 146–160, 2005.
143. Bernard P. Zeigler. *Theory of Modelling and Simulation*. Krieger Publishing Co., Inc., 1984.

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188		
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing this collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.					
1. REPORT DATE (DD-MM-YYYY) 12-09-2019		2. REPORT TYPE Doctoral Dissertation		3. DATES COVERED (From - To) Aug 2016 - Sep 2019	
4. TITLE AND SUBTITLE Emergent Behavior Development and Control in Multi-Agent Systems			5a. CONTRACT NUMBER		
			5b. GRANT NUMBER		
			5c. PROGRAM ELEMENT NUMBER		
6. AUTHOR(S) King, David W., Major, USAF			5d. PROJECT NUMBER		
			5e. TASK NUMBER		
			5f. WORK UNIT NUMBER		
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Air Force Institute of Technology Graduate School of Engineering and Management (AFIT/EN) 2950 Hobson Way WPAFB, OH 45433-7765			8. PERFORMING ORGANIZATION REPORT NUMBER AFIT-ENG-DS-19-S-007		
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) Major Garrett E. Duff Studies, Analyses and Assessments 1720 Air Force Pentagon Washington D.C. 20330-1570			10. SPONSOR/MONITOR'S ACRONYM(S)		
			11. SPONSOR/MONITOR'S REPORT NUMBER(S)		
12. DISTRIBUTION / AVAILABILITY STATEMENT APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED					
13. SUPPLEMENTARY NOTES This work is declared a work of the U.S. Government and is not subject to copyright protection in the United States.					
14. ABSTRACT Emergence in natural systems is the development of complex behaviors that result from the aggregation of simple agent-to-agent and agent-to-environment interactions. Emergence research intersects with many disciplines such as physics, biology, and ecology and provides a theoretical framework for investigating how order appears to spontaneously arise in complex adaptive systems. In biological systems, emergent behaviors allow simple agents to collectively accomplish multiple tasks in highly dynamic environments; ensuring system survival. These systems all display similar properties: self-organized hierarchies, robustness, adaptability, and decentralized task execution. However, current algorithmic approaches merely present theoretical models without showing how these models actually create hierarchical, emergent systems. To fill this research gap, this dissertation presents an algorithm based on entropy and speciation - defined as morphological or physiological differences in a population - that results in hierarchical emergent phenomena in multi-agent systems. Results show that speciation creates system hierarchies composed of goal-aligned entities, i.e. niches. As niche actions aggregate into more complex behaviors, more levels emerge within the system hierarchy, eventually resulting in a system that can meet multiple tasks and is robust to environmental changes. Speciation provides a powerful tool for creating goal-aligned, decentralized systems that are inherently robust and adaptable, meeting the scalability demands of current, multi-agent system design. Results in base defense, k-n assignment, division of labor and resource competition experiments, show that speciated populations create hierarchical self-organized systems, meet multiple tasks and are more robust to environmental change than non-speciated populations.					
15. SUBJECT TERMS Multi-agent systems, emergence, speciation, decentralized systems					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT UU	18. NUMBER OF PAGES 194	19a. NAME OF RESPONSIBLE PERSON Dr. Gilbert L. Peterson
a. REPORT U	b. ABSTRACT U	c. THIS PAGE U			19b. TELEPHONE NUMBER (include area code) (937) 255-3636 x4281