3-2022

Performance of Heterogeneous Multi-Agent Systems with Applications in Combined Arms

Robert J. Wilson

Performance of Heterogeneous Multi-Agent Systems with Applications to Combined Arms

THESIS

Robert J Wilson, Captain, USAF

AFIT-ENG-MS-22-M-074

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT-ENG-MS-22-M-074

PERFORMANCE OF HETEROGENEOUS MULTI-AGENT SYSTEMS WITH APPLICATIONS TO COMBINED ARMS

THESIS

Presented to the Faculty

Department of Electrical and Computer Engineering

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 Master of Science in Cyber Operations

Robert J Wilson, B.S.M., B.A.T.

Captain, USAF

March 19, 2022

DISTRIBUTION STATEMENT A
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

AFIT-ENG-MS-22-M-074

PERFORMANCE OF HETEROGENEOUS MULTI-AGENT SYSTEMS WITH APPLICATIONS TO COMBINED ARMS

THESIS

Robert J Wilson, B.S.M., B.A.T.
Captain, USAF

Committee Membership:

Lt Col David W. King, Ph.D
Chair

Gilbert L. Peterson, Ph.D
Member

Lt Col Wayne C. Henry, Ph.D
Member

Barry E. Mullins, Ph.D, P.E.
Member

AFIT-ENG-MS-22-M-074

Abstract

Multi-agent systems show great potential for solving problems in complex and dynamic domains. Such systems comprise multiple individual entities called agents. The overall behavior of the system emerges from the many interactions of its component agents. The majority of studied systems comprise homogeneous agents, which possess the same behavior or physical form. However, recent work suggests that heterogeneous agents, which possess diverse behaviors or forms, may improve system performance. This research examines the impact of heterogeneity on multi-agent system effectiveness and investigates the application of multi-agent systems to combined arms warfare, which simultaneously applies heterogeneous unit types to accomplish military objectives. Hundreds of morphologically homogeneous and heterogeneous multi-agent teams were evolved and evaluated on their ability to complete certain objectives. Results indicate that no single team configuration excels in all scenarios, and that the ability to shift between heterogeneous and homogeneous configurations is more important to team success than any configuration's heterogeneity. Results further show that combined arms tactics, as described in United States Marine Corps doctrine, can emerge from the interactions of simple, decentralized agents, indicating that future research in this domain may prove valuable to the military art of combined arms warfare.

For Dad

Table of Contents

Page

Page

List of Figures

Figure | Page

List of Tables

PERFORMANCE OF HETEROGENEOUS MULTI-AGENT SYSTEMS WITH APPLICATIONS TO COMBINED ARMS

I. Introduction

1.1 Problem Background

Modern combined arms doctrine calls for the simultaneous application of multiple armament types to achieve an effect greater than the sum of its parts [1, 2]. This phraseology is familiar to researchers of Complex Adaptive Systems (CASs) as the language of emergence, a phenomenon by which a system of relatively simple agents exhibit complex aggregate behaviors and thus create 'much from little' - an effect greater than its summed parts [3, 4, 5]. John Boyd, on whose work current combined arms theory is based, saw armed forces as CASs comprising complex networks of autonomous units in various roles and drew on CAS literature to develop his theories of warfare [6]. For Boyd, every military unit fills a specialized niche that contributes to the operations of the whole, so that the development of an effective combined arms force is a multi-agent systems engineering problem. In such a problem, each agent must be designed such that by interacting with its fellows it helps produce some desirable trait at the system level. Victory in battle is a desirable trait for an army, for example, and the training and equipment of each individual soldier is directed to overall task of winning battles.

A combined arms force is specifically a heterogeneous multi-agent system. The term *heterogeneous* denotes differences between agents in morphology, in behavior, or both. *Morphological* heterogeneity denotes differences in physical characteristics while

behavioral heterogeneity indicates differences in the way agents act on sensed data. Consequently morphologically heterogeneous agents differ as tanks from airplanes or police dogs from policemen. Behaviorally heterogeneous agents differ as two identical aircraft might perform different roles during a mission, or as one soldier might provide covering fire while another advances [3, 7].

The history of combined arms warfare offers many examples of heterogeneous systems, from combined formations of skirmishers, infantry, and cavalry in antiquity to composite wing concepts in the modern Air Force [8, 9]. Modern computer simulations and research in Artificial Intelligence (AI) offer new opportunities to advance combined arms theory by modeling and evaluating unit compositions and tactics. Recent research demonstrates agents capable of generating novel tactics in strategy games [10], coordinating multiple physically and behaviorally distinct units to perform collaborative tasks in the physical world [11], and generating novel and diverse behaviors in cooperative teams of agents [12]. All of these results are directly relatable to combined arms theory. Such research furthers both the military arts and the study of AI and multi-agent systems.

1.2 Research Questions

It is hypothesized that a heterogeneous system of agents will accomplish assigned tasks more effectively than homogeneous systems, where effectiveness is measured by a fitness score measuring successful task completion. More specifically, this research will answer the following questions:

1. Does *behavioral heterogeneity* improve or impair the performance of multi-agent systems in a combined arms scenario?

2. Does *morphological heterogeneity* improve or impair the performance of multi-agent systems in a combined arms scenario?

3. Given a set of morphologically distinct units, can a multi-agent system exhibit synergistic combined arms behavior without explicit central direction?

Questions one and two deal with the ways in which diversity can be introduced to groups of agents. Is it beneficial to employ agents with physical or morphological differences? To answer these questions, the present research tested and compared behaviorally and morphologically heterogeneous teams in a variety of different tasks. The third question assesses the potential of multi-agent systems to engage in the types of cooperative behavior seen in combined arms warfare.

This research presents several simulations of heterogeneous teams of fighting units and evaluates the effect of both morphological and behavioral heterogeneity on team effectiveness. Multiple heterogeneous and homogeneous teams were generated and tested in four scenarios, each with distinct objectives. Teams were graded according to their rate of victory and in inverse proportion to damage taken. The results were used to identify the most effective and efficient team configurations and behaviors.

The fittest teams exhibited cooperative tactics, including flanking maneuvers, scouting, multi-pronged attacks, and other behaviors. These tactics emerged from the interactions of each team's member agents and often combined disparate agent morphologies or behaviors. This research demonstrates that the effect of heterogeneity on team fitness varies according to the situation, that the most effective teams tend to evolve heterogeneous behaviors and morphologies to overcome tactical challenges, and that combined arms tactics can emerge from the interactions of simple agents.

1.3 Contributions

This research provides a comparison of homogeneous and heterogeneous multi-agent systems in a dynamic domain. It supports Department of Defense (DoD)

priorities for the development of autonomous weapon systems [13] and applies multi-agent systems theory to the military art of combined arms by demonstrating the emergence of recognizable tactical behavior from simple agent interactions. A new and extensible simulator is provided for future research on single- and multi-agent systems.

1.4 Outline

Chapter II provides relevant background and research on multi-agent systems and outlines the field's relationship to combined arms theory. A description of the RoboCodePlus simulator is also supplied. Chapter III describes the methodology used to perform experiments outlining agent architectures, test scenarios, and the genetic algorithm used to generate and evolve individual teams. Chapter IV analyzes the results of each experiment and draws conclusions, while Chapter V summarizes the work done and provides recommendations for future efforts.

II. Background and Related Work

The advancement of computing technology is closely accompanied by the development of Artificial Intelligence (AI) theories and their applications. Section 2.1 attempts to answer two questions. First, what is an agent? Second, what makes an agent or a system of agents intelligent? Sections 2.2-2.4 discuss different types and examples of multi-agent systems. Section 2.5 describes contemporary multi-agent teaming experiments in real-time games. Section 2.6 relates the field of heterogeneous multi-agent systems with the military concept of combined arms warfare. The final section introduces the simulator used to conduct experiments.

2.1 Intelligent Agents

John Holland adapted the term *agent* from economics to denote active elements in a system [14]. Agents are defined by boundaries, where a boundary is a semipermeable barrier that accepts certain signals and blocks others [15]. Within the boundary is some system, describable as a set of rules, which converts collected stimuli (the permeating signals) into appropriate action. Agents often contain multiple boundaries so that one agent may itself be a system of many smaller agents, called an *aggregate agent*. Consider, for example, a human being made up of many organic cells. Each cell is an agent, defined by a boundary and acting according to chemical signals from other cells. The human being is an aggregate agent made up of many cells, bounded in three-dimensional space and reacting to such signals as observed light, sound, social cues, etc. Holland's definition is easily generalized and can refer to agents in biology, computer science, economics, or any other domain with active elements.

In the realm of computer science, Russell and Norvig define an agent as anything which perceives and acts upon its environment using sensors and actuators [16]. When

one considers sensors as determinants for which signals may permeate a boundary, this definition aligns with Holland's lexicon. Russell and Norvig further define a *rational* agent as one which acts so as to maximize some measure of performance [16]. In other words, for an agent to be *rational* it must act so as to transform its current state into some desired future state. This matches Franklin's and Graesser's definition of an autonomous agent as a system which acts on its environment over time in pursuit of an agenda [17]. By conceiving of an agent as a system in itself they echo Holland's vision of aggregate agents. But what makes such a system intelligent?

The question is difficult to answer because no consensus exists on what constitutes intelligence. Turing famously proposed a game by which a machine's intelligence might be measured by the similarity between its behavior and a human's [18]. Taking a similar line, Minsky characterized the problem of creating AI as designing "machines capable of performing tasks that would require intelligence if done by humans" [19]. In distinguishing an intelligent agent from a rational one, Russell and Norvig merely write that a rational agent might be considered intelligent if it is successful [16] - if, in other words, the agent performs assigned tasks with a high probability of success. These four, Turing, Minsky, Russell and Norvig, sidestep the question of what intelligence *is* to instead ask what it *does*.

Effective computing agents have been devised for a number of tasks that seem to require intelligence in humans, with many famous examples in the domain of games. In 1959, Arthur Samuel devised a checkers-playing computer program capable of learning from previous games, eventually outperforming its human programmer [20]. Forty years later, IBM's Deep Blue made headlines by defeating Gary Kasparov in a chess match [21]. Early versions of the Go-playing AlphaGo agent developed by Deepmind proved capable of defeating the strongest known human Go players [22]. Its successor, AlphaGo Zero, was able to train itself to even higher levels of expertise with

no prior information outside the game's rules [23]. Whether any of these systems was intelligent in any meaningful sense is a matter of debate, but they do meet Minsky's benchmark by performing tasks associated with human acuity.

Dissatisfied with classical conceptions of symbolic AI, Brooks argued that to display intelligence, systems must have their representations grounded in the physical world [24]. *Embodied intelligence*, as it came to be called, is observable in organisms that tailor their actions to their environments based on sensory cues [25, 26] and Brooks contended that it is prerequisite to the symbolic intelligence required for abstract tasks like playing chess. He therefore proposed a subsumption architecture for robotics control that grouped processes into independent behaviors triggered by environmental stimuli[1]. Robots in Brooks' experiments exhibited goal-seeking behaviors emerging from this simple architecture. The principle of subsumption is further developed in [27], where robots filter external inputs through a hierarchy of state-driven behaviors to select the appropriate action for a given state.

Much writing on the nature and measurement of intelligence exist, and Chollet's excellent work on past, present, and future theory in general AI is recommended [28]. This research does not aim to produce *intelligent* agents or even *intelligent* teams, having no complete definition of that term's meaning. This research aims to evaluate the performance of multi-agent systems in light of certain variables - specifically the heterogeneity of its components. To develop such systems it follows Brooks' example, employing simple agents with behaviors tied to sensory cues. In the spirit of Turing, the resulting systems are judged by how effectively they perform given tasks, while the question of intelligence is left to the philosophers.

[1]Brooks behavior-oriented architecture strongly resembles the rule-driven decision frameworks in Holland's agents[14].

2.2 Multi-Agent Systems

Shoham and Leyton-Brown define multi-agent systems as “those systems that include multiple autonomous entities with either diverging information or diverging interests, or both” [29]. Each agent in such a system makes its own observations and determines what actions it will take [16]. A multi-agent system is a Complex Adaptive System (CAS) if it possesses many components and a capacity to adapt to environmental changes [30].

The process by which aggregate interactions of relatively simple agents produce complex system behaviors is known as emergence [4, 3]. Thus the manifold interactions of individual cells drive the behaviors of biological organisms, the interactions of billions of neurons give rise to cognition, and the many independent actions of bees in a swarm make up a hive. A fundamental challenge of engineering multi-agent systems, particularly CAS, is dictating the behaviors of individual agents in such a way that a desired system-level behavior emerges.

Multi-agent systems can offer a number of advantages over single-agent solutions. Many distributed agents are able to process their own sensory inputs and act without waiting on a centralized arbiter. Failure of a single agent is usually not catastrophic to a system and in a behaviorally adaptive system a surviving agent can change its role to cover for an important disabled agent [31]. Experiments by van Lon and Holvoet showed that multi-agent solutions outperform centralized algorithms in logistical problems exhibiting high dynamism, high urgency, and large scale [32]. Dynamism refers to a domain’s rate of change, where a continuously changing domain is highly dynamic [33]. Urgency measures the time a system has to generate responses to signals and high urgency means less available time. The scale of a problem refers to the number of factors that must be taken into account. Their results suggest that multi-agent systems are well-suited to real-time scenarios in complex domains.

Many examples of multi-agent systems exist in nature, and multi-agent software systems are frequently modeled after the behaviors of biological organisms. Packs of wolves present one such example. Wolf packs in the wild display a number of cooperative hunting strategies, such as ambushing and relay running. Some researchers theorized that these behaviors might indicate high-order intelligence functions like planning and foresight in wolves [34]. Muro, et al. demonstrate, however, that wolves' hunting strategies are reproduced by multi-agent systems of software agents in which each simulated 'wolf' follows two simple rules [35]:

1. Move towards prey until at a minimum safe distance
2. Once at minimum safe distance, move away from other wolves close to the prey (see Figure 1)

Their wolf agents exhibited apparent ambushing and relay hunting behaviors in hunt simulations, demonstrating the emergence of intelligent pack behavior from very simple individual interactions. Muro, et al.'s, work shows that high-level, complex behaviors do not require complicated component rules or agents. Their work suggests that one can create CAS from relatively simple agents.

Figure 1: A pack of wolves surrounds a lone bison in the final stage of a hunt [35]

2.3 Systems of Homogeneous Agents

Muro, et al.'s wolf pack model is an example of a homogeneous multi-agent system. A multi-agent system is homogeneous if all agents possess the same attributes [36, 37, 3]. Muro's wolf agents are homogeneous in two important respects. They are *morphologically* homogeneous because they possess the same physical characteristics: the same physical speed, stamina, etc. They are *behaviorally* homogeneous because they follow the same set of behavioral rules. Bonabeau, et al. in 1999 and Dorigo, et al. in 2013 noted that a preponderance of prior research on multi-agent systems focused on systems of homogeneous agents [38, 11]. The wolf pack algorithm above is just one example from a large collection of homogeneous multi-agent systems. Ant Colony Optimization (ACO) and Particle Swarm Optimization (PSO) are two of the most widely known homogeneous multi-agent system algorithms.

Dorigo, et al. first proposed the Ant System [39] and later developed the ACO algorithm [40]. The basis for the system came from observations of ant colonies and the signals exchanged between individual ants. When traveling from the nest to a food source and back, each ant deposits a small amount of pheromone along its path [41]. Other ants will follow this path with probability proportional to the amount of deposited pheromone, and since each ant leaves its own pheromone trail, a more heavily traveled path attracts more ants. Shorter paths are preferred because ants arrive at the destination and return more quickly, depositing the same amount of pheromone in a shorter time. Pheromone evaporates over time so that paths not under continual use will eventually cease to attract new travelers. From the many such interactions among the ants of a colony emerges a sophisticated pathfinding behavior that helps the colony find the shortest paths between food sources and the nest.

ACO variants show very good performance on Hamiltonian path finding problems

such as Knight's Tour[2] [42] and Traveling Salesman [43]. Generalized forms of ACO can be applied in cyber intrusion detection [44] and continuous optimization problems [38].

Craig Reynolds developed the boid model to simulate the motion of groups of birds, fish, or ungulates by having agents follow three rules [45]:

1. Avoid collisions with fellow agents

2. Try to match nearby agents' velocity

3. Try to stay close to nearby agents

Reynolds referred to his agents as 'boids' (for 'bird-oid objects'). By following these simple behaviors, boids exhibited complex flocking behaviors similar to those observed in flocks of birds or schools of fish. Similarly, James Kennedy and Russell Eberhart leveraged Reynold's work to develop PSO, a successful application of the boid paradigm used to train neural networks and optimize nonlinear functions by imitating the way flocks navigate to sources of food [46]. Many variants of PSO have since been developed, sometimes unwittingly. Supposedly novel algorithms like Grey Wolf Optimizer (GWO) or the Firefly algorithm, both with many applications and citations in current research, were shown to be variants of PSO by Villalón, et al. [47]
.

Agents in both PSO and ACO are very simple, determining behavior based on their immediate surroundings. In PSO, each boid adjusts its attitude according to its surrounding agents, while in ACO an ant chooses its path based on pheromone trails which indicate the popularity of a given path with other agents. In each instance, the system's function (pathfinding, optimization, etc.) is performed only by all agents

[2] A knight's tour is a sequence of moves that would allow a chess knight to traverse every square on a chess board without repetition.

operating in cooperation and could not be accomplished by any single ant or boid. In each system too, only one type of agent is employed. An ant or a boid in either algorithm differs from other agents only in temporary parameters such as position, thus both systems are homogeneous in their classic forms.

In nature, groups of organisms are seldom so uniform. Most ant colonies exhibit division of labor on lines based on differences in morphology. Leafcutter ants will recruit larger workers - called ‘majors’ - to defend the nest in case of attack [48] and workers of certain species of army ant also have a distinct major class with larger mandibles and a stinger, specialized for combat [49]. Wolves exhibit different behaviors based on age, sex, and such variable factors as the time of mating season or age of offspring [50]. These natural systems are, in contrast with their artificial imitators, composed of heterogeneous rather than homogeneous agents.

2.4 Systems of Heterogeneous Agents

Agents in a heterogeneous system possess differing attributes. Heterogeneity may be morphological or behavioral [7]. Morphological heterogeneity refers to differences in physical form, as exists between a tank and a staff car, or between a worker and a soldier ant. Behaviorally heterogeneous agents may be morphologically homogeneous, but employ different logic to determine which actions to take and may therefore behave differently. King refers to behavioral heterogeneity as *physiological speciation* and notes that this form of variation tends to allow more flexible behaviors than morphological heterogeneity [3]. This is presumably because physical differences often cannot be varied during task execution, but behaviors can be changed according to circumstance.

Just before the new millennium, Bonabeau, et al. noted a dearth of research focused on designing heterogeneous systems [38]. Twenty years later, homogeneous

systems continue to command the bulk of attention in such fields as Swarm Intelligence (SI) [51], but the body of work for heterogeneous systems has expanded. Several heterogeneous variants of PSO have emerged and are outlined in [52]. Each variant displays heterogeneity in a different agent attribute: neighborhood size, selection mechanisms, update rules, or update parameters. Dorigo, et al. designed the Swarmanoid system, which employs morphologically distinct and mutually complementary robot types in a heterogeneous swarm [11]. A simplistic overview of the three robot types in Swarmanoid are as follows:

1. Eye-bots: flying robots with cameras able to get an aerial view of the environment

2. Hand-bots: climbing robots unable to move on the ground, but with the ability to climb vertical structures and retrieve small objects

3. Foot-bots: modular tracked and wheeled robots able to move around on the ground, with an LED system for signalling other robots

Swarmanoid demonstrated its viability by exploring an environment and retrieving an object from a shelf without prior knowledge of the test area.

Kengyel, et al. show that swarms with heterogeneous behaviors can outperform homogeneous swarms in two-dimensional goal-finding tasks [7]. King explores techniques for engineering morphologically and behaviorally heterogeneous swarms in [3] and notes that the two classes of speciation, morphological and physiological/behavioral, have different effects on the system's performance. Deka and Sycara show natural development of heterogeneous strategies among morphologically homogeneous agents during a competitive reinforcement learning scenario [12]. Additionally there is a large body of literature on the performance of heterogeneous agents in real-time games.

2.5 Heterogeneous Agents in Real-Time Games

Games like *Chess* or *Go* are turn-based and sequential. This means that each game consists of a series of discrete game states represented by the positions of pieces on the board. With the move: '1. e4', a game of *Chess* transitions from its starting state, with white to move, to a modified state with black to move, and so it goes for the rest of the game. In other turn-based games the turns are taken simultaneously. In the game *Diplomacy*, players reveal their moves at the same time and determine the resulting game state by calculating the result of their aggregate decisions [53].

Real-time games take place in continuous time. Due to the complexities of simulating a domain in which all elements are updated continuously and simultaneously, real-time games are typically simulated on a computer[3] or acted out by live agents, as in a military exercise or a sport. Real-time games possess higher dynamism than turn-based games, based on the criteria laid out by van Lon, et al. [33]. They take place in a continuous environment and can involve large numbers of distinct units with specialized behaviors. The potential state space for such games is tremendous, with even simple real-time scenarios exceeding the state space of board games like *Chess* by factors of 50 or more [55]. The complexity and dynamism of such games makes them good test beds for multi-agent systems [56].

Multi-agent systems have been developed to play a variety of real-time games and simulations[4]. Many of these systems exhibit morphological or behavioral heterogeneity, or both. Three such examples are found in pursuit-evasion style games in which a

[3]Computer simulations of real-time games possess the interesting quirk that they are not technically continuous. A real-time strategy game played on a machine must be executed by discrete cycles of the computer processor, and most game engines will update game state in time steps that update the positions of objects in the game based on a 'good enough' approximation of in-game physics rules [54]. Given that many games will execute thirty, sixty, or even more such updates in a second to keep up with monitor refresh rates, the resulting state spaces are enormous and may for all intents and purposes be considered continuous.

[4]For purposes of this research, no distinction is drawn between the terms 'game' and 'simulation'. Both create an environment in which agents operate according to a set of rules.

set of morphologically identical 'pursuit' agents attempt to capture similarly uniform 'evader' agents. Comparing flocking strategies for ghosts in the game Ms. Pac-Man, in which the roles of pursuer and evader are occasionally temporarily reversed, Liberatore, et al. found homogeneous teams following a shared strategy were more effective than heterogeneous teams in which each ghost could select from multiple strategies [57]. Deka and Sycara found that morphologically homogeneous agents in their Fort-Attack simulation naturally learned heterogeneous strategies and that these strategies increased their teams' chances of success [12]. In one such example, attacking agents employed most of the team as decoys, distracting defender agents while one attacker sneaked past to the goal. King, et al. used the Informal Task Assignment Algorithm (ITAA) to assign different roles to morphologically homogeneous agents in a similar nest-defense scenario [58], where flanking maneuvers spontaneously emerged from the combination of agent-held roles and signalling.

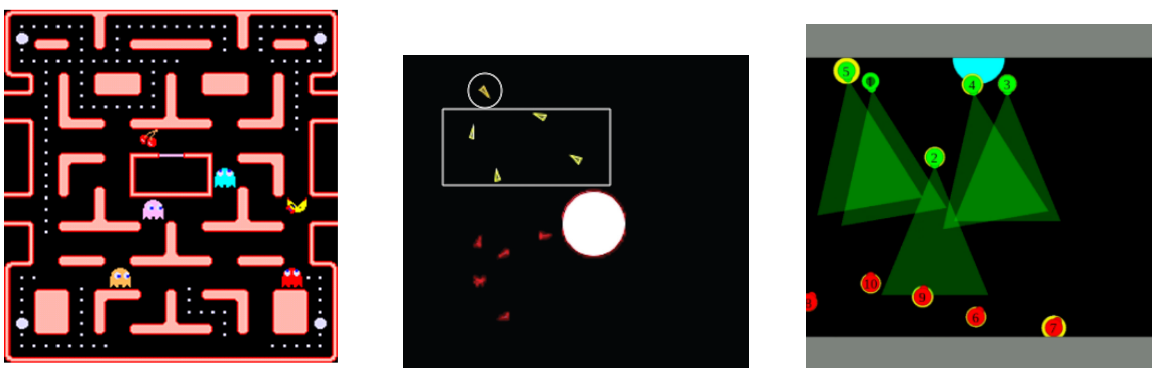

Figure 2: Screenshots from Ms. Pac-Man (left) [59], King's nest defense game (center) [60], and Deka's FortAttack game (right) [12]. In Ms. Pac-Man the player (yellow circle) attempts to avoid the ghosts while collecting the white dots. In King's and Deka's games, evader agents attempt to reach the target zone (large white circle) while pursuit agents attempt to intercept them. All three are types of pursuit-evasion game.

In the StarCraft: Brood War and StarCraft II Real-Time Strategy (RTS) games players control a variety of morphologically heterogeneous units. They use these units to collect resources, construct buildings and additional units, and engage in combat.

While the objective of the game depends on the game mode, it is most commonly to destroy all units owned by competing players. StarCraft's enduring popularity, wide recognition, and well-documented Application Programming Interface (API)s [61, 62] have made the games popular as platforms for AI development and testing.

The EISBot StarCraft player developed by Weber, et al. is composed of collections of behaviors called 'managers', each of which handles a different aspect of the game, from worker assignment and resource collection to reconnaissance and combat [56]. Weber's system dynamically assigned heterogeneous behaviors to game units based on in-game conditions, but was highly centralized. By contrast, in the Swarm-GAP system implemented in StarCraft: Brood War by Tavares, et al., heterogeneous agents select tasks from a global pool according to each agent's capabilities and situation in the game [63]. Tavares, et al. include a 'commander' agent type that conducts logistical tasks like deciding what units to build.

Adhikari took a slightly different approach by co-evolving agent behaviors for individual fighting units in StarCraft II [64]. His research focused on a subdomain of the overall game - namely combat between groups of units - and left out aspects like resource gathering. Adhikari's model distributed agent control to the individual. Each unit was represented by its own agent, and while each agent shared overall objectives: conservation of ally health, destruction of the enemy, etc. It chose actions independently of any central control mechanism. These distributed agents were able to defeat mid-tier human StarCraft II players in simple group combat scenarios. In a similar vein, the StarCraft Multi-Agent Challenge (SMAC) developed by Samvelyan, et al. lays out a set of challenges in which each StarCraft unit is controlled by an independent agent [65]. Teams in SMAC must learn to cooperate in partially observable environments in order to defeat teams of enemy agents.

Any treatment of StarCraft II agents must consider the premier specimen de-

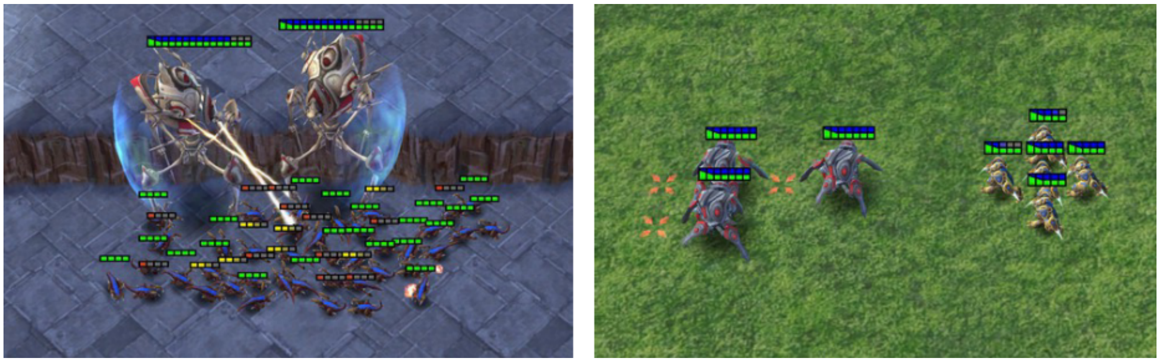

Figure 3: SMAC scenarios presented in [65]. While the scenarios shown involve competing teams of morphologically homogeneous units, other SMAC scenarios involve heterogeneous teams.

veloped at Deepmind, the organization that produced AlphaGo. The imaginatively named AlphaStar agent consistently defeats human StarCraft II players at the Grandmaster level, the highest tier in the StarCraft II player ranking scheme [66]. AlphaStar is a centrally-controlled architecture that uses reinforcement learning and domain knowledge extracted from replays of games by the top 20% of human players. The most recent version is subject to many of the same limitations as a human player - Actions per Minute (APM) are limited to ensure the agent does not operate at speeds far beyond what a human could achieve, and the agent's view of the map is limited to the same limited single-pane screen view a human player sees on the computer monitor [10]. AlphaStar is therefore a centralized agent which controls multiple subordinate agents, i.e. the in-game units. The diverse unit types and complex domain of StarCraft and other RTS games make them good candidates for experiments in a vein of military theory called *combined arms.*

2.6 Heterogeneous Agents for Combined Arms Operations

During operations in Quang Nam province in the Vietnam war, 3d Battalion, 26th Marines under Lt. Col John Studt contended with an asymmetric enemy that favored

ambush tactics over direct confrontation. To cope with this threat, the Marines began taking dogs on patrol, and Studt credits to this practice the fact that not a Marine in his battalion was ambushed over the course of months of operations in the area [67]. Studt's pairing of Marines with dogs makes a classic case study of advantages in heterogeneous agent systems. The dog is a dedicated sensing agent, able to use its superior sensitivity to smells and sounds to warn the Marines - fighting agents - of enemy teams they might not otherwise detect. If the team were made up of only Marines, it would be less able to sense and respond to hidden threats. An all-dog unit would entail even more severe limitations. The combination of dog with Marine yields a higher chance of success than an all-dog or all-Marine unit.

Such heterogeneity is a staple of military operational art throughout history. Alexander the Great defeated the Persian Empire with combined forces of cavalry, skirmishers, and heavy infantry [8]. During the American Civil War, General Grant relied on coordinated action by naval and ground forces to attain his objectives in the Vicksburg campaign and when taking forts Henry and Donelson [68]. World War I saw such innovations as the creeping barrage and combined assaults by mutually supporting forces of infantry, artillery, armour, and aircraft [69, 70]. The *Blitzkrieg* concept employed by Germany in World War II used the inherent characteristics of airpower, artillery, and mechanized units to break up enemy systems into isolated centers of gravity [6], a process equivalent to destroying the communication links between agents in a multi-agent system. The Air Force has several times experimented with 'composite wings' of heterogeneous aircraft types [9].

Combined arms doctrine as practiced by the U.S. Army and Marine Corps is "the full integration of arms in such a way that to counteract one, the enemy must become more vulnerable to another" [2]. It combines heterogeneous units such as aircraft, artillery, infantry, etc., to achieve effects greater than might result from

their separate or sequential application [1]. The employment of combined arms is therefore a problem of division of labor in a heterogeneous multi-agent system. In such problems, as in combined arms maneuvers, the objective is to determine the optimal distribution of simultaneous tasks to a population of agents [71].

Many of the dilemmas encountered by agents in StarCraft and other RTS games are essentially combined arms conundrums. What is the optimal ratio of unit types in a group? How can a heterogeneous team of units be used to their greatest effect? Consequently a review of the military history and doctrine of combined arms is a fruitful vein of inquiry for heterogeneous agent research, and vice versa. The AlphaStar agent mentioned in Section 2.5 evolved a number of novel unit combinations and tactics not hitherto seen in human play. Similar experiments might unearth novel approaches to combined arms.

Examples of existing combined arms tactics can be found in United States Marine Corps (USMC) doctrine and theory. Marine Corps Doctrinal Publication (MCDP) 1 outlines the synergistic pairing of automatic weapons with grenade launchers as an example of combined arms. Suppressing fire from an automatic weapon keeps an adversary stationary and makes him vulnerable to grenades, while if he attempts to avoid grenades by changing position he is exposed to the automatic weapon [2]. In a more generalized example, Lind suggests Marine rifle squads split into probing and support teams in the field. The probing team aggressively probes enemy positions for weaknesses while the support team provides suppressing fire [72]. Any attempt to engage the probing team will expose the enemy to the support team. The former example displays morphological speciation via technology. The second can be performed by a homogeneous group of riflemen split into teams with heterogeneous behaviors (probe and support), but is more effective if the different groups are equipped according to their special roles - thus Lind recommends automatic weapons for the support

team.

Figure 4: Multi-agent systems at different levels of military organization
Left: USMC fire team wedge [73]
Center: Assets of a USAF composite wing [74]
Right: Structure of an Army combined arms battalion [75]

It is reasonable to expect that further research into systems of heterogeneous agents may yield benefits for military operational art in any of several ways:

1. Supporting the development of autonomous and mutually supporting weapon systems
2. Designing autonomous weapon systems that complement and integrate with existing assets
3. Suggesting effective asset combinations for a given mission type
4. Identifying ways to degrade, disrupt, or destroy enemy multi-agent systems

Real-time games provide a good testing ground for combined arms concepts because they allow for simulated combat at an accelerated pace without the loss of life associated with hard-earned experience in actual conflict. The next section describes one such game, RoboCodePlus, which enables experiments with teaming of different unit types and could be used to model combined arms scenarios.

2.7 The RoboCodePlus Platform

RoboCodePlus is a novel simulation built to test multi-agent teams of agents. It is based heavily on RoboCode [76], a real-time computer game in which robotic vehicles duel for supremacy, but is written in C++ and differs from RoboCode in several respects.

2.7.1 RoboCode

The original RoboCode is a real-time battle royale game in which tank-like robots fight in free-for-all or team engagements. RoboCode was originally released by IBM in 2001 [77] and used to teach computer science AI programming concepts. Robots in the game are morphologically homogeneous and consist of three components - the body, the turret, and the radar. The body simulates a tracked chassis and provides motive power and orientation. The turret and radar can swivel independently of the body. The turret fires energy bullets to damage other bots and the radar detects bot positions. Robot visibility is limited such that a robot's radar is the only way for it to gather information on its opponents.

Games take place over multiple rounds. The robot that wins the most rounds, wins the game. Each robot starts a round with 100 energy. Firing the gun costs energy in proportion to the power of the bullet (robots can dynamically select bullet power at firing time). A robot's energy is also reduced if it collides with a bullet or another robot. If a robot depletes its energy reserves by firing it is disabled and becomes unable to take any action. If the robot's energy is reduced to 0 or less by a collision the robot is destroyed. Energy is increased whenever the robot successfully hits another robot with a bullet. Every round, scores are assigned to each robot on the basis of damage dealt and relative longevity.

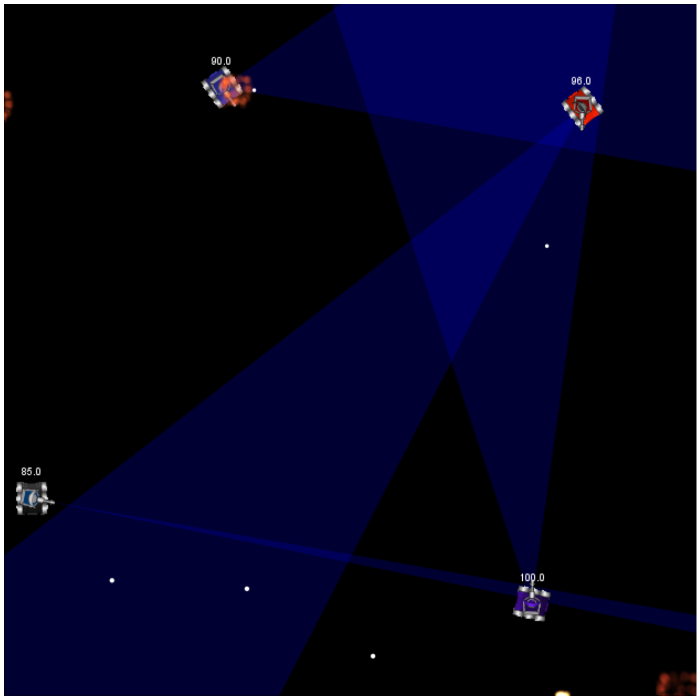

Figure 5: Screenshot from the original RoboCode, with sensor fields visible.

2.7.2 Existing Research in RoboCode

Woolley and Peterson demonstrate a Unified Behavior Framework (UBF) for abstracting modular high-level behaviors like 'Wander' and 'Shoot' from low-level controls for simulated RoboCode entities or real-world robots [27]. Multiple researchers wrote genetic algorithms capable of evolving competitive, machine-written RoboCode agents [78, 79, 80]. Rebelo, et al. developed a team model for RoboCode in which agents are behaviorally heterogeneous according to emotional models. The team's course of action is selected based on a voting scheme in which each bot submits its suggestion to a leader and the leader makes a decision based on the aggregate votes and an *esteem* score assigned to each robot [81].

Recchia, et al. used a modified version of RoboCode to conduct experiments particularly relevant to the current research [31]. In these experiments each robot was subdivided into distinct commander, gunner, and driver agents. These agents

cooperated to identify and target hostile bots while navigating the battlefield and coordinating with friendly robots. Agents in the same robot did not communicate directly except in that commander, gunner, and driver each had access to all elements of a robot's state. Inter-robot communication was very limited. Robots notified team members of their current targets, and emitted a distress broadcast when they suffered large amounts of damage from a particular enemy. Each team comprised five robots including one designated leader. The leader had more energy and a radar, while the other four robots lacked radar and had to rely on communications from the leader to determine the positions of targets. The behaviors of driver, gunner, and commander agents were varied across tests but were held constant across teams so that each team always exhibited a homogeneous set of controlling agents. Consequently teams possessed morphological heterogeneity, i.e. the leader's enhanced attributes, but were behaviorally homogeneous at the level of the robot.

Recchia, et al.'s results showed that teams with cooperative, team-oriented behaviors tended to outperform teams of agents with more selfish, individually-focused behaviors [31]. Agents were termed selfish if they chose actions based only on their own context, like targeting whatever enemy was nearest or had last damaged the agent. Cooperative agents instead selected actions based on the context of the team, such as coordinating attacks with allies so that multiple teammates focus on the same robot at once.

2.7.3 RoboCodePlus

The experiments above are a promising first step in exploring heterogeneous teaming with RoboCode, but suffer from platform constraints. In the current version of RoboCode there is no way to implement morphologically heterogeneous agents without heavily editing the engine source code, and team composition is limited to a

narrow set of possible configurations. To provide this and other functionality, and to pave the way for heterogeneous systems research, RoboCodePlus was written from the ground up to retain the general structure of RoboCode while implementing additional features:

1. Where RoboCode's agents were morphologically homogeneous, consisting of identical tanks with body, gun and sensor. RoboCodePlus incorporates morphologically distinct agents with differing characteristics and allows customization of individual components, from tank tracks and weapon turrets to sensors and ammunition.

2. In RoboCode each robot is scored based on damage dealt, enemies destroyed, and time survived. RoboCodePlus introduces objective-based scenarios in which a team attains victory by meeting a particular condition, such as reaching a designated zone or destroying one particular enemy.

3. The box2d physics engine is used to manage in-world collisions, creating a more rich simulation and more opportunities for agents to discover novel, unexpected tactics.

2.7.4 Robots

As in RoboCode, the key units of RoboCodePlus are the robots themselves. Each robot is composed of a behavior and a set of components. Components are loosely categorized as impellers (things like tracks or tires used for motive power), weapons, and sensors, but the component interface is sufficiently modular that arbitrary component types can be introduced. Each component possesses a set of associated actions and each action is associated with an integer-mapped set of control states. A simple turret, for example, can rotate left or right, fire, or change ammunition types. These

states can be mapped as shown in Table 1.

Table 1: Simple Turret Control States

Control State	**Turret Action**
0x01	Rotate clockwise
0x02	Rotate counter-clockwise
0x04	Fire
0x08	Load ammunition type 1
0x10	Load ammunition type 2

Behaviors follow the Unified Behavior Framework interface outlined in [27]. Each robot is associated with a Behavior interface through which it is assigned actions for a given time step, with high-level actions like 'Rotate turret θ radians per second and fire' mapping to the control states for a given bot structure. Researchers can develop their own implementations of the Behavior interface. These can be simple, static behaviors, e.g. spin and shoot until energy is depleted, or more complex composite behaviors, e.g. patrol the map counter-clockwise and scan the area until an enemy is detected, then move towards the enemy and open fire.

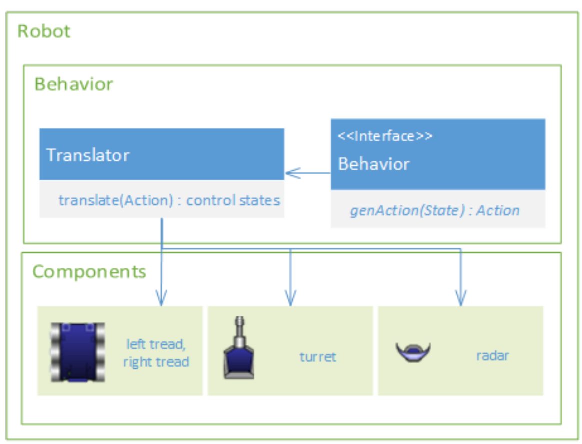

Figure 6: Basic structure of a simple robot in RoboCodePlus. Behaviors generate actions, which are translated into control states for execution by the components.

A team of robots in RoboCodePlus is *behaviorally heterogeneous* if the robots' Behavior implementations differ. It is *morphologically heterogeneous* if its members

differ in their armament, components, or in characteristics such as energy levels or maximum speed. RoboCodePlus consequently provides a suitable engine for comparing the effects of both types of heterogeneity on team performance in a range of tasks.

2.8 Summary

This chapter reviewed definitions of intelligent agents and explored existing research on multi-agent systems. Particular attention was paid to real-time game agents and to connections between multi-agent systems theory and the military doctrine of combined arms. The penultimate section introduced the RoboCodePlus framework and reviewed existing work which utilized the old RoboCode platform. The next chapter will introduce a set of experiments used to compare homogeneous and heterogeneous teams of robots in RoboCodePlus.

III. Methodology

3.1 Research Goals

The purpose of this research is to test the hypothesis stated in Chapter I; namely, that heterogeneous systems of agents will perform assigned tasks more efficiently than homogeneous systems. Heterogeneity may be present in morphological or behavioral characteristics, as differentiated in Chapter II. Consequently two types of experiments were carried out. The first tested the effect of behavioral heterogeneity in a team of morphologically homogeneous robots. This was accomplished by using a genetic algorithm to evolve the teams' behaviors and evaluating the tactics of the fittest teams. The second experimental set added morphological variance by evolving robots' physical templates along with their behaviors. Robots mutated between three possible variants: Tank, Scout, or Artillery.

3.2 Robots

A robot in RoboCodePlus is an autonomous unit linking a Behavior with a map of Components. A team of robots is therefore morphologically heterogeneous if they differ in their components and behaviorally heterogeneous if the robots have different behaviors. A particular set of morphological configurations, dubbed 'templates', are used in these experiments.

3.2.1 Templates

A template is a particular combination of components and parameters that describe a robot's physical composition - a morphological archetype. These experiments use four basic robot templates: Tank, Artillery, Scout, and Turret.

- **Tank:** A typical robot comparable to those in the original RoboCode, with a simple gun and radar.

- **Artillery:** A slower, long-range robot which launches exploding munitions and does not have a sensor. Firing rate is limited to 1.2 rounds per second (r/s) as opposed to the tank's 3 r/s. Upon detonation, artillery shells explode and project 32 rays in a tight circular pattern, inflicting 5 damage per intersecting ray to any robot caught in the blast. Artillery shells travel very slowly compared with other munitions. Artillery robots have no sensor and must rely on team members to spot targets.

- **Scout:** A small and fast-moving bot with less energy than the other templates and no weapon. Its advantages are increased speed, a smaller profile, and an extended sensor range.

- **Static Turret:** A stationary robot with twice the energy of a tank and no means of movement; used to defend static positions and not included in morphological evolution. The muzzle velocity of turret projectiles is significantly higher than that of robots, making evasion difficult.

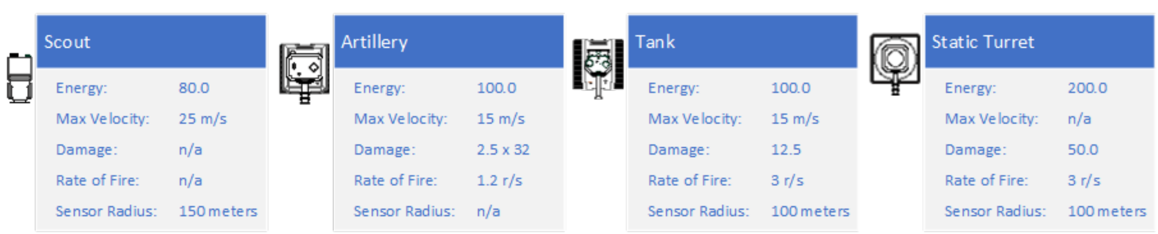

Figure 7: Robot morphologies and associated properties

3.3 Team Structure

Teams employed in this exercise are decentralized. This means that each agent chooses its actions independently, without direction from a controlling unit. Coop-

erative behaviors instead emerge from interactions between agents - either by the exchange of messages or by the collection of sensor data. Each robot can send and receive simple messages containing useful information, such as the locations of detected enemies or an action the robot intends to take in the future. Team members can then factor these messages into their decision-making process.

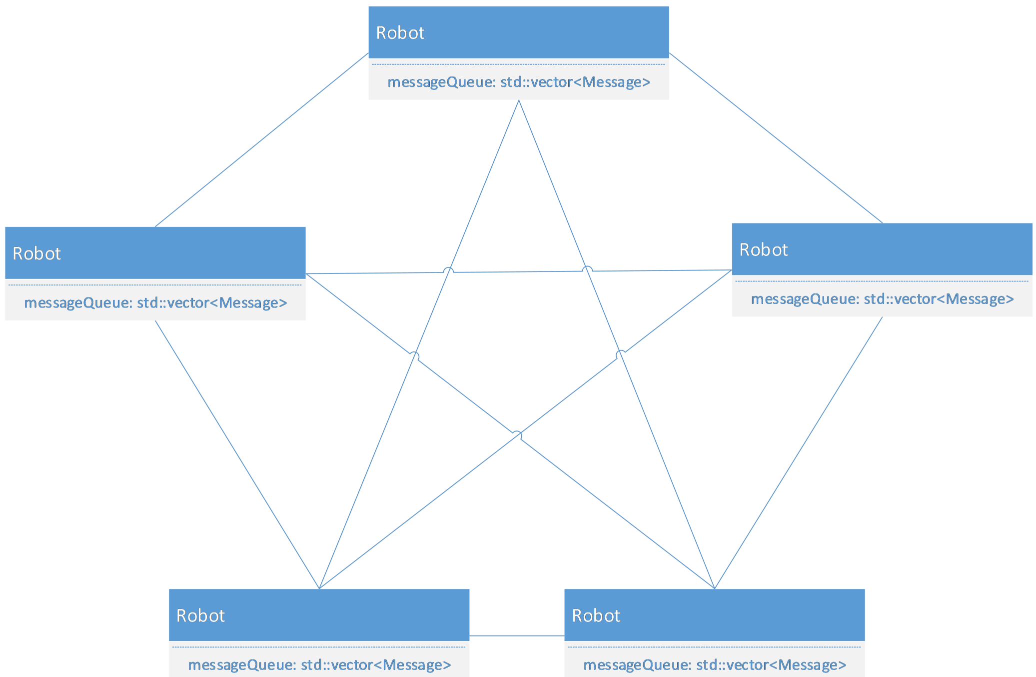

Figure 8: Each agent in a decentralized team can send messages directly to any other team mate, and there is no centralized controller

3.3.1 Messages

Team members communicate via messages, which consist of a 'from' field, a 'to' field, a set of bit flags, and a data section. The first two fields specify the message's sender and intended recipient, while the bit flags indicate a message's type. The type indicates a message's intended purpose, which may be prescriptive, such as 'target this robot' or 'move to this point', or purely informational, as 'energy level critical'. Robots generate and exchange messages to request action by other robots or to update

team members as to their status or current course of action. Creating and sending appropriate messages is part of the process of action generation.

3.4 Scenarios

Each scenario defines starting positions and objectives for two teams. The first team to complete all of its assigned objectives wins the scenario. Teams were evaluated on their performance in each scenario, and their overall performance across the aggregate set of scenarios. Effective team strategies vary based on the scenario objective, so that the best team for a zone defense scenario may perform poorly in an elimination scenario, and vice versa. The tested scenarios are labeled zone attack, zone defense, elimination, and last team standing. Each scenario has a different objective and creates a different challenge for the evolving team.

3.4.1 Zone Attack

Objective: Get at least one allied robot into the designated zone.

The goal is to reach a target zone with at least one team member. The size and placement of the zone are identical to the layout in zone defense, but in this scenario the zone is defended by static turret robots. The turrets are more durable than a normal robot but cannot move, except to rotate their weapons. The crux of this scenario is effectively overcoming static defenses while sustaining as little damage as possible. The turrets' superior durability and damage gives the defenders an inherent advantage, and only the attacking team is evolved for the Zone Attack scenarios.

3.4.2 Zone Defense

Objective: Do not allow any enemy robots to reach the designated zone.

Zone defense is a ‘keep-away’ scenario in which the team attempts to prevent an adversary team from entering the protected zone. The defended zone is a circular area ten meters in radius, placed at the coordinates (100,200) on a 200x200 meter battlefield. The attacking team is composed of two scout robots which have no weapons but move more quickly than tanks or artillery. The defending team of five robots has a significant numerical advantage, but this is not always sufficient for victory. Effective teams must adapt to intercept the elusive scouts as consistently as possible.

3.4.3 Elimination

Objective: Destroy a particular robot.

In an elimination scenario, the goal is to destroy one particular robot on the opposing team. The enemy team is composed of five tanks, including the target and four defenders who will attempt to interpose themselves between the target and any detected attackers, while firing on the nearest adversary. The attacking team must circumvent the defenders and destroy the target robot as quickly as possible. The defenders must protect the target for 25 seconds of game time. The defending team possesses a numerical advantage.

3.4.4 Last Team Standing

Objective: Destroy all robots on the opposing team

Both teams contain equal numbers of robots. A victorious team must therefore create an imbalance which allows it to inflict more casualties than it takes. If the game continues for more than 84 seconds in game time, the match is declared a draw.

3.5 Agent Control

Robots are controlled by Behaviors which generate actions based on the robot's current state. Several heuristics may be applied to determine the desirability of a particular action. Different behaviors may be more or less effective depending on the scenario objectives and on particulars of the robot state.

The Unified Behavior Framework outlined by Woolley and Peterson uses a variety of arbitration strategies to construct the best action for a given situation from a set of simple behaviors [27]. This research adopts the Unified Behavior Framework (UBF) and combines the elemental behaviors developed by Woolley and Peterson with new behaviors geared towards teamed agents. Here called *atomic* behaviors, they serve as building blocks for more complex decision-making processes.

3.5.1 Atomic Behaviors

Each atomic behavior is a simple directive: move to this point, scan this arc, move towards allies, etc. The general categories of atomic behaviors used in these experiments are as follows[1]:

1. **Wander:** Move in pseudo-random manner, randomly selecting a new heading and velocity after a given number of turns.

2. **Patrol:** Move around the map perimeter in a clockwise or counter-clockwise direction.

3. **Charge:** Move as quickly as possible towards a given point on the battlefield. The target point may be static (reach this particular section of terrain) or mobile (chase this robot's last detected position).

[1] Several of these behaviors, such as *Track*, are derived directly from behaviors designed by Woolley and Peterson [27]

4. **Disperse:** Move away from team's center of mass.

5. **Contract:** Move towards the team's center of mass.

6. **Flee:** Move away from the nearest enemy.

7. **Fire:** Fire weapon at a given heading. Multiple Fire behaviors might be incorporated in creating a single action - one for each weapon component attached to the robot.

8. **Scan:** Rotate sensor across a designated arc. Once again this behavior may be be repeatedly applied, with each application corresponding to one of multiple sensors.

9. **Track:** Tracks a designated target by rotating turret and sensor to match bearing to a designated target.

10. **Relay Target:** Alert team members to location of a scanned robot.

Some of the descriptions above encompass several actual atomic behaviors. *Patrol*, for example, covers the behaviors *Patrol*, *ReversePatrol*, and *RandomPatrol*. The first behavior causes a robot to circle the perimeter in a clockwise direction, the second to travel counterclockwise, and the third to vary the direction at the start of the game or when struck by a munition. A complete listing of atomic behaviors may be found in Appendix A.

3.5.2 Behavior Heuristics

Composite behaviors construct actions from atomic behaviors based on the robot's state at the end of each turn. Atomic behaviors become active or modify their effect based on details of the current state. For example the *RandomPatrol* behavior causes a robot to alternate between a clockwise or counterclockwise route around the map

when it is struck by a munition. *TargetWeakest* only activates if the robot detects or is notified of at least one enemy, and designates as a target the detected enemy with the least remaining energy. These and other heuristics for behavior construction may be drawn from particulars of the game state, yielding effective reactions to occurrences on the battlefield. A non-exhaustive list of such heuristics is given below.

3.5.2.1 Proximity to Team Members

Different team formations may be more or less effective, depending on the scenario. A dispersed formation may be able to overwhelm static defenses, while a tighter formation may better shield a protected team member. The distance from the robot to its team's center of mass - the average of the positions of every team member - helps robots to judge their positioning relative to the rest of the team.

3.5.2.2 Target Position

The closer a robot is to a target, the sooner it can reach the target (especially important for zone defense/control) and the more difficult it will be for the target to avoid the robot's fire. This heuristic is useful in Elimination or Zone scenarios where the objective is to destroy a particular robot or reach a particular section of the map.

3.5.2.3 Proximity to Nearest Threat

Distance to the nearest enemy robot may prompt a fight response in a tank with little or no damage, or a flight response in an unarmed scout. A damaged robot in close proximity to an enemy may send a message to team mates requesting help.

3.5.2.4 Threatened Team Member

A family of heuristics may be based upon messages received from teammates. One such heuristic encourages the robot to move towards a damaged team mate, and potentially to interpose itself between the team member and the nearest threat.

3.5.3 Action Selection

A simple priority fusion arbiter along the lines of the colony architectures described by Woolley and Peterson [27] is used to combine atomic behaviors into more complex composite behaviors for each robot. Each composite behavior comprises four atomic behaviors with associated weight values. The weight determines a given behavior's priority. If multiple atomic behaviors define different actions for a particular state, then the behavior with the higher weight takes priority.

An example is shown in Figure 9. The robot shown has four atomic behaviors: *Wander*, *TargetClosest*, *ChargeRobot*, and *Scan*. The weights are shown in green boxes to the right of each behavior. Since *TargetClosest* is the only behavior controlling the gun and turret and *Scan* is the only behavior controlling the sensor, these will always be the operative behaviors for controlling those components of the robot. Both *Wander* and *ChargeRobot* control the robot's turn rate and velocity, but the former is only active if and when a particular enemy is detected. Consequently the robot will wander until the target is detected, at which point *ChargeRobot* will cause the robot to accelerate towards the target.

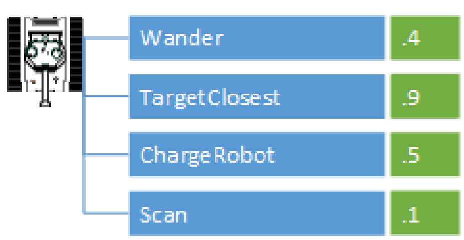

Figure 9: A composite behavior with four components

3.5.4 The Behavioral Problem Space

A composite behavior of the type used in this research can comprise any of c^A potential combinations of atomic behaviors, where c is the number of component behaviors and A is the number of possible atomic behaviors. For example, 21 atomic behaviors were used in these experiments, and each composite behavior has four component behaviors. This allows for $21^4 = 194,481$ potential variations, disregarding weights. In a heterogeneous team three possible morphologies expand the search space for a given robot to $3 \cdot 21^4 = 583,443$. Consequently the total search space for a single team is is $\left(3 \cdot c^A\right)^n$, where n is the number of robots on the team.

3.6 Team Evaluation

A team consists of a particular combination of robot templates and behaviors. Teams are evaluated based on three metrics: their win rate W and conservation C. Win rate is a simple average of the number of victories v over the total number of rounds r for a given scenario, with $W = 1$ representing a perfect winning record Equation (1). Conservation C stands for the proportion of robots on the team which survived the round, and the proportion of energy conserved. Teams that accomplish their objectives with fewer robots and less energy lost are rewarded. C is set to 0 in any round that the team lost[2]. A composite of W and C yields a composite fitness score F Equation (3). F was used to rank teams from most to least effective.

$$W = \frac{victories}{rounds} \tag{1}$$

[2]Otherwise teams find local optima in which all team members sat at their starting locations, since this behavior minimized sustained damage

$$C = \begin{cases} .5\left(\frac{survivingRobots}{totalRobots}\right) + .5\left(\frac{remainingEnergy}{startingEnergy}\right) & \text{if team won round} \\ 0 & \text{if team lost round} \end{cases} \tag{2}$$

$$F = .6(W) + .4(C) \tag{3}$$

Separate W, C, and F scores are calculated for each scenario and may be labeled with a subscript. W_{ZD}, for example, would denote a team's win rate in the Zone Defense scenario. An aggregate average win rate across all scenarios is denoted by a simple W with no subscript: $W = \frac{W_{ZD}+W_{ZA}+W_E+W_S}{4}$, and likewise for C and F.

3.7 Evolution

Algorithm 1 is used to generate and evolve new teams based on their fitness scores. It operates on a population P of 100 robots, generating the initial population at random. At each iteration the algorithm appraises each team in the population in ten simulations of the current scenario. It selects the ten best performing teams - the elites - from P and discards the rest. It then uses the elites to generate new

Algorithm 1 Run Experiment

```
function runExperiment(Population P, Scenario S, Iterations I)
    for i ← 1, I do
        ε ← (I−i)/(2·I)                         ▷ Update exploration variable
        evolve(P)
        resetScores(P)                          ▷ Clear fitness from previous runs
        for p ∈ P do                            ▷ Iterate individual teams in population
            reset(S)
            for j ← 1, 10 do
                playRound(S)                    ▷ Play 10 rounds per team
            end for
            addCredit(p)                        ▷ Calculate fitness
        end for
    end for
end function
```

Algorithm 2 Evolve Population

```
function EVOLVE(Population P)
    SORT(P)                                  ▷ Sort population by fitness
    n ← ELITES                               ▷ Set number of elites
    for i ← n, P.count do
        r ← RANDOM(0, 1)                     ▷ Get random number in [0,1]
        if r < ε then
            RANDOMIZE(P[i])
        else
            p1, p2 ← GETRANDOMELITE()
            P[i] ← CROSSOVER(p1, p2)
        end if
    end for
end function
```

teams via crossover. As illustrated in Figure 10, the crossover process selects child team behaviors and weights at random from each parent team. Random mutations may also occur, with a 5% probability of mutation for each behavior. In Figure 10, the fourth behavior, *TargetWeakest*, of the child's tank robot is not found in either parent, but occurred through random mutation. If morphological mutation is part of the current experiment it also occurs in this step with a 5% probability, otherwise the child retains the first parent's morphologies. The resulting team is then inserted into the population.

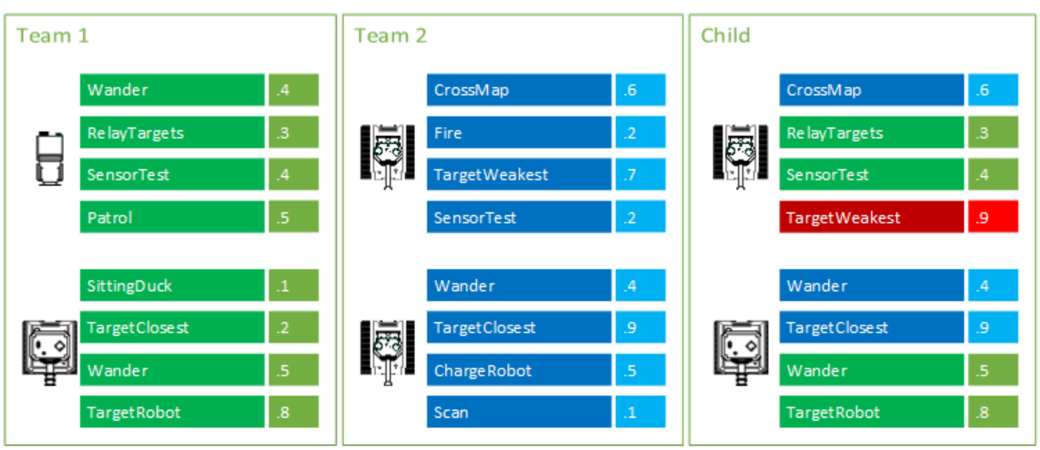

Figure 10: Crossover between two morphologically heterogeneous teams of two robots

The number of new teams generated via crossover is determined by the variable ε, which decreases as the iteration count increases. ε designates the fraction of the population to be generated randomly rather than by crossover and hence balances the priorities of exploration and exploitation of solutions. It is largest at the start of an experiment, when half of the population is generated randomly at a pass, and smallest as the experiment draws to a close. Consequently, early iterations favor exploration, the discovery of new team configurations via random generation. Later iterations focus on exploitation, generating the majority of new teams by crossover between elites in the current population.

Opposing teams in the scenario take turns evolving. Once one team has completed 100 iterations of evolution, the opposing team evolves for a hundred iterations, and so on. Thus once one team has developed a successful winning strategy, the other team is evolved to counter that strategy, and so on[3]. The initial team is evolved six times and the opposing team five, for a total of eleven rounds of evolution per experiment. The adversarial approach produces a developing sequence of tactics and counter-tactics.

3.8 Environment

All scenarios take place on a 200x200 meter battlefield, where meters are in-game physics units. Movement is constrained to the space within the battlefield. Physics are deterministic for a given scenario configuration but the starting positions of at least one team are randomized in each scenario, so the configuration varies for every game round. All simulations are conducted on a computer system driven by a 3.5 GHz Intel Core i7-5930k processor, with 24 GB 2133 MHz DDR4 RAM, running the Windows 10 Operating System (version 19042).

[3]The Zone Attack scenario is alone excepted, as only the attacking team is evolved.

3.9 Summary

This chapter outlined the general methodologies used for experiments in the course of this research. The first set of experiments is designed to test the effect of behavioral heterogeneity on teams of agents. The second also evaluates the effects of morphological heterogeneity. This section also described the composition of robots, behaviors, and teams in RoboCodePlus and the methods and functions used to evaluate teams. The next chapter will describe each experiment in detail and analyze its results.

IV. Results and Analysis

This chapter presents results for the experiments described in Chapter III. Section 4.1 presents overviews of the tactics and counter-tactics which emerged from the experiments. Sections 4.2 through 4.5 describe the experiments themselves and provide figures which characterize the behavior of each team after each round of evolution. Each scenario was tested with homogeneous and heterogeneous teams. Tested teams displayed a number of cooperative behaviors, including flanking, multi-pronged attacks, and scouting. Section 4.6 describes another set of experimental iterations in which the relative performance of morphologically heterogeneous and homogeneous teams are compared, and Section 4.7 concludes this chapter with a summary of findings.

Each team's approach to an objective emerges from the collective behaviors of its members. Therefore when a team is said to be performing a 'flank' maneuver it does not mean robots are being directed to take that route by some central controller. Rather it means that the flanking robots have evolved behaviors which cause them to move in the same direction as an approximate group. It may be that all of the flanking robots employ the 'Patrol' atomic behavior, or that most members are patrolling while others attempt to converge on the team's center of mass, causing them to follow the group. This distinction should be kept in mind when reading the accounts below.

4.1 Emergent Tactics

During the simulation, contending teams adopted different macro-level behaviors according to the situation at hand. From these behaviors recurring themes may be extracted. The fact that these themes were most clearly evident when the evolving team had a clearly defined objective beyond destroying all opponents - 'destroy this

robot' for example, or 'get to this zone' - is noteworthy.

4.1.1 Flanking

Flanking behavior emerged in at least some iterations of most scenarios. A flanking movement occurs when a group of attackers strikes at a target's flank, where the flank is "the right or left limit of a unit" [82]. Flanking attacks force defenders to rotate to meet the oncoming attack. In these scenarios, flanking maneuvers typically struck at a smaller face of the defending formation than would have met a frontal attack. The defensive formation in the large-team Elimination scenarios presented a front of six or seven robots, for example, while the Eastern or Southern face presented two or three. Flanking attacks allow the attacking team to apply forces against a limited number of defenders and make it more difficult for defenders on the opposite end of the formation to engage effectively. In most cases the flanking maneuver also created an enfilade, described in Section 4.1.3.

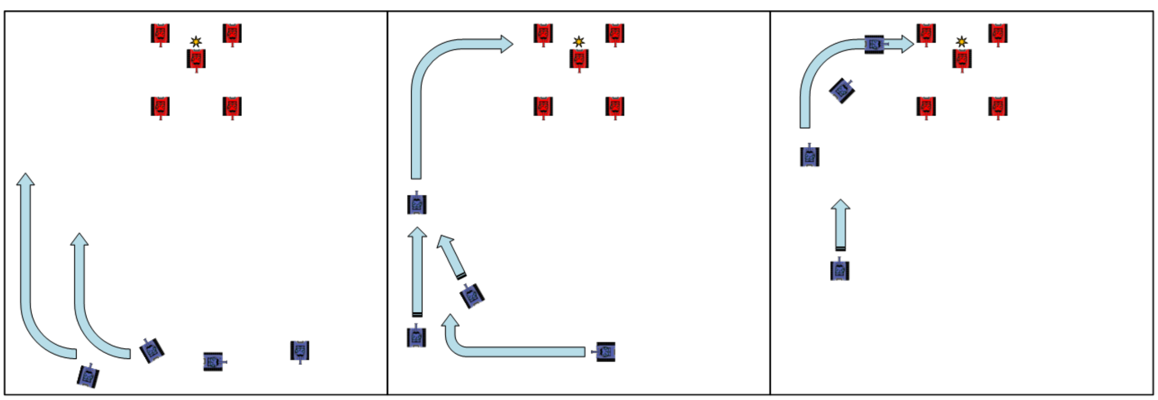

Figure 11: Attacking team (blue) performs flanking maneuver in an Elimination scenario. Not to scale.

The response to flanking maneuvers tended to vary based on the objective type. In Elimination scenarios the solution was usually to flank in the opposite direction, often causing evolutionary oscillation as teams switched flanking directions to catch

or preserve the target robot. In Last Team Standing scenarios, it was common to see centrally located artillery firing at flanking units as they passed. The artillery shells' explosive force tended to throw off the flankers' maneuvering and to savage close formations.

4.1.2 Multi-Pronged Attacks

Multi-pronged attacks are attacks in which distinct groups of robots take distinctly separate approaches to the same destination. They emerged frequently when the attacking team was faced with formidable defenses around a defined target: examples include the target zone guarded by turrets in the Zone Attack scenario, or the target robot in the Elimination scenarios. In Elimination scenarios, the behavior emerged specifically when defenders formed a tight cluster around the target, when multiple approaches maximized the attackers' chance of slipping an attack through the defensive huddle. They also arose regularly in Last Team Standing scenarios, often allowing the attackers to attack knots of enemies from multiple angles.

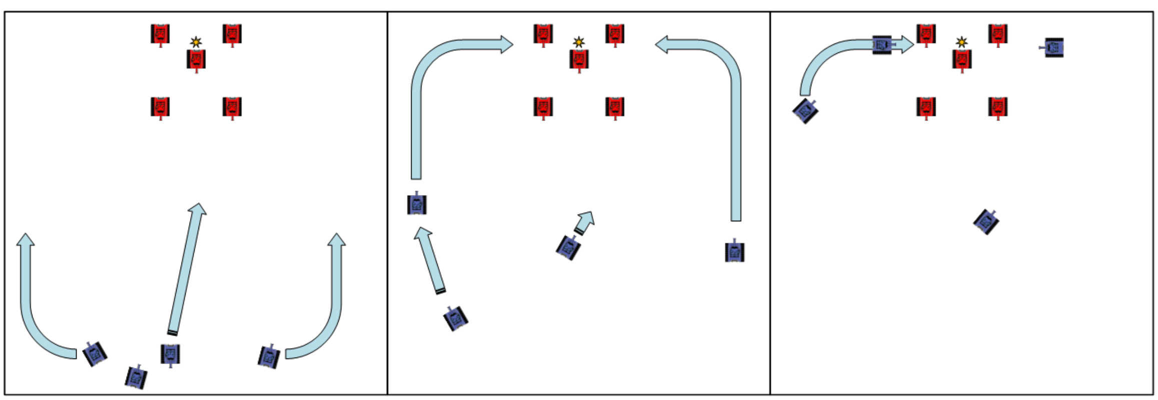

Figure 12: Attacking team (pale blue) executes a multi-pronged attack in elimination scenario. Right and left flanks close in a pincer while the central robot stands off.

4.1.3 Enfilade

According to the United States Marine Corps (USMC), enfilading fire is "delivered so that the long axis of the beaten zone coincides... with the long axis of the target" [73]. Such an attack has the advantage of potentially striking an enemy even if it falls short or long of the intended target. It has the added advantage that defenders often interfere with each others' ability to return fire. In Figure 13 tank A is enfilading the red formation by firing along its long axis. Only the leftmost red tank can return fire from its current position. By contrast, tank B is approaching from the formation's front and can be targeted by all three red tanks at once. Enfilading fire emerged in several experiments, usually following a flanking maneuver, as in the first evolution of the attacking team in the homogeneous Elimination scenario. In that instance the bulk of the attacking team flanked East, then turned and attacked West along the long axis of the defending team.

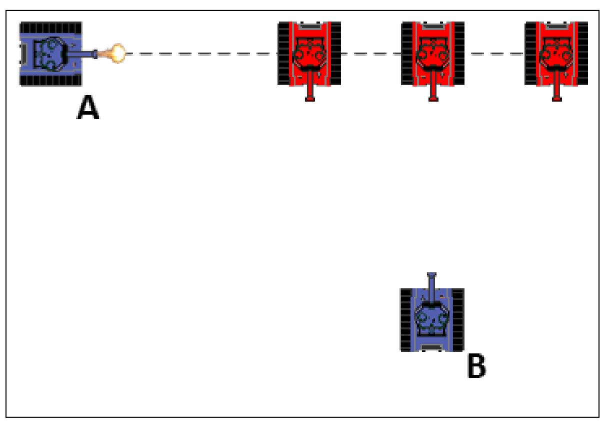

Figure 13: Tank A is enfilading the red formation by firing along the long axis

4.1.4 Cooperative Scouting

In morphologically heterogeneous teams, scouts frequently evolved composite behaviors including sensor sweeps, some kind of wandering or patrolling movement, and

the *RelayTargets* atomic behavior. The scouts identified targets for tanks or artillery (see Figure 14). In the latter case, this symbiotic relationship was a necessity born of morphology. Having no sensors themselves, the artillery robots relied on tanks or scouts to identify targets. Scouts also assisted tanks by extending the range of tanks' detection. Tanks alerted by scouts would rotate turrets to anticipate an enemy's arrival, for example, or maneuver towards enemies out of the tanks' own sensor range.

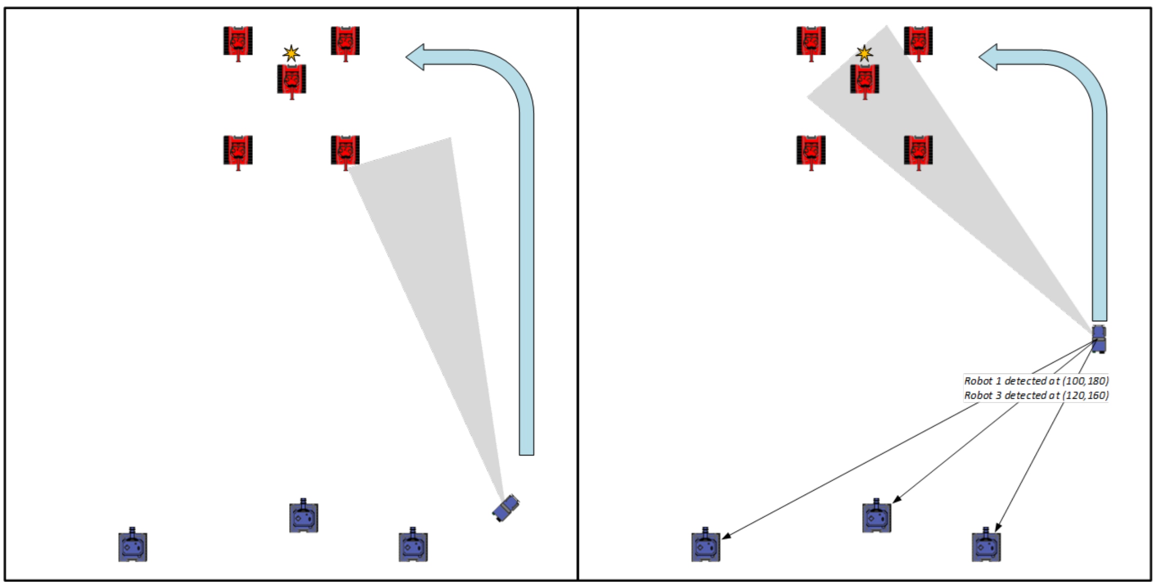

Figure 14: A scout identifies targets for artillery in an Elimination scenario. Artillery robots have no sensors and rely on scouts or tanks to spot targets for them.

4.1.5 Decoying

In the final round of morphologically heterogeneous evolution for the Elimination scenario, the three scouts on the attacking team performed two functions. The first was to identify targets for allied artillery. The second was to serve as bait. Moving rapidly to the center of the map, the scouts were quickly identified and targeted by defenders as the closest detected enemies. The attacking artillery was consequently

left unmolested long enough to destroy the target (see Figure 15). A similar tactic emerged on the opposite team when defenders entangled attacking robots while the elimination target fled.

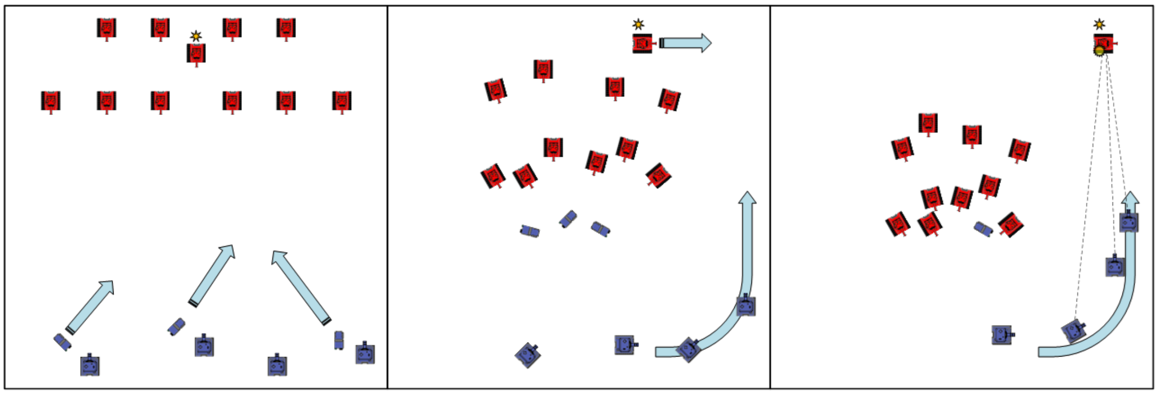

Figure 15: Blue team's scouts draw defenders while artillery fires on the target.

4.1.6 Ambush or Interception

In cases where the route of a fleeing target was known, robots of the attacking team often moved to intercept by traversing the same route in reverse, or simply waited for the target to come to them. Likewise defenders sometimes moved to preemptively intercept and engage attacking forces when protecting a robot or a zone (see Figure 16). The static nature of the atomic behaviors, which generally adopt a set direction for movement, made interception a common and fruitful tactic.

4.1.7 Summary

The sections above describe various cooperative tactics which emerged during experimentation. The next four sections outline the results of each experiment set. Each section is divided into three subsections. The first is an account of tactics evolved by morphologically homogeneous teams, the second an account of tactics and compositions evolved by morphologically heterogeneous teams and the final section is a

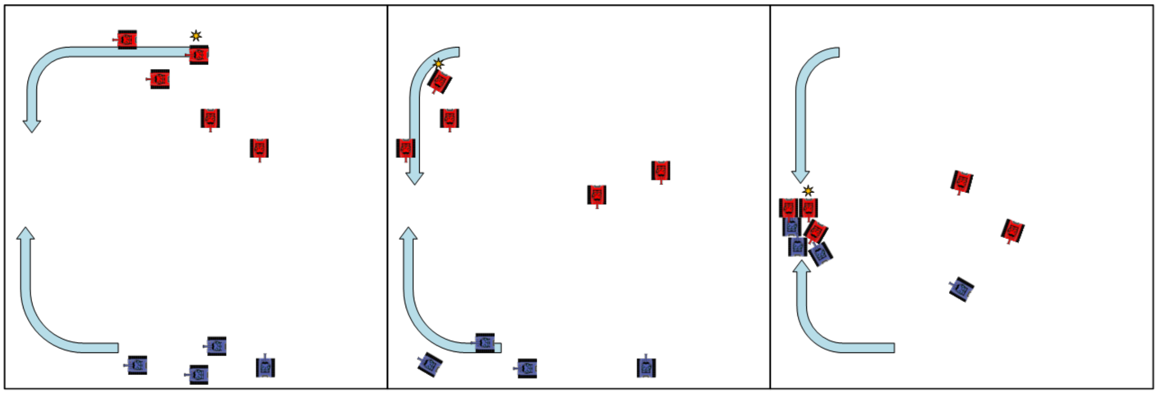

Figure 16: Blue team entangles the fleeing target by placing robots on its route of escape.

summary. Section 4.2 contains an additional subsection immediately before the summary, comparing the performance of behaviorally and morphologically homogeneous control teams with that of teams from the first two sections.

4.2 Scenario One: Zone Attack

In Zone Attack scenarios the defending team consists of four static turrets, each of which possess twice the energy of a regular tank robot and four times the damage output. Each static turret possesses a sensor which detects any robot within 100 meters and follows the *TargetClosest* atomic behavior, causing it to fire on the closest detected enemy. A single static turret is capable of destroying any other robot type with two successful hits. The goal of the attacking team is to get within 30 meters of the point (100,200), shown in Figure 17 as a green circle. The attacking team consists of five robots placed randomly within 20 meters of the battlefield's South edge. Only the attacking team evolves in Zone Attack.

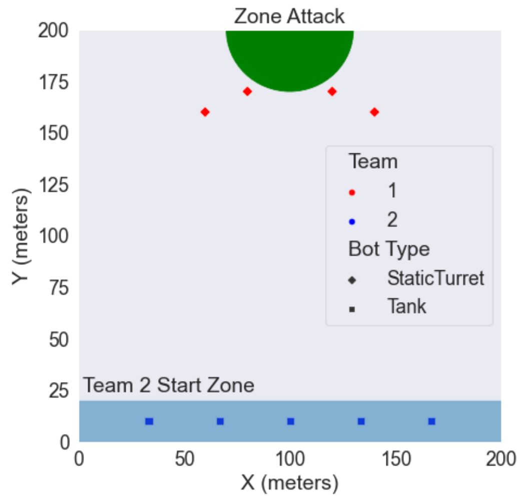

Figure 17: Zone Attack starting positions

4.2.1 Morphologically Homogeneous Teams

Across six rounds of morphologically homogeneous evolution, the attacking team developed two effective tactics for reaching the target zone (see Table 2). The first was a multi-pronged approach in which one to two tanks moved to the zone around the West flank, one to two approached from the East, and one to two approached directly down the center (see Figure 12). The second was a variation of the multi-pronged approach in which one or more robots served as decoys, loitering in the center of the map and drawing turret fire while flanking units moved to the target zone. Both behaviors are examples of heterogeneous tactics, with different robots adopting different roles to help the team reach its objective.

Four of the six rounds of homogeneous evolution incorporated a robot which used the *Concentrate* atomic behavior to coordinate a central approach. *Concentrate* causes a robot to approach its team's center of mass. When team mates are flanking to the East and West the robot with *Concentrate* will try to keep between them. A robot with the *ChargePosition* or *CrossMap* behaviors crosses the center of the map, often rushing ahead of flankers going the long way around. A robot with

Concentrate, by contrast, keeps pace with the flankers so that all team members come within range of the adversary turrets at approximately the same time. Turrets have to deal with all of the attackers at once, rather than destroying them one by one. This tactic therefore fulfills the combined arms principle of simultaneous action, by which arms are synchronized "to achieve an effect greater than if each element was used separately or sequentially" [1].

Table 2: Tactics of the six fittest morphologically homogeneous teams in the Zone Attack scenario.

Iteration	Tactics	Fitness
1	Three-pronged attack, synchronized	.081
2	Three-pronged attack, synchronized, with decoy	.066
3	Three-pronged attack, with decoy	.094
4	Three-pronged attack, synchronized	.093
5	Three-pronged attack	.092
6	Three-pronged attack, synchronized	.094

4.2.2 Morphologically Heterogeneous Teams

Morphologically heterogeneous teams favored largely homogeneous morphologies during evolution, favoring scout robots. The scout, with its small profile and higher maximum speed, is well suited to slip past static defenses. The six fittest heterogeneous teams employed four scouts and one tank. They evolved the same tactics as homogeneous teams, with three-pronged attacks, decoys, and synchronized approaches (see Section 4.2.2). An additional, homogeneous tactic emerged wherein groups of scouts simply rushed the target zone en masse, each scout taking the straightest route from its starting position to the target. This simple tactic, impractical for groups of slow-moving tanks, proved viable for small, fast scouts. This approach emerged in one out of the six fittest morphologically heterogeneous teams.

Table 3: Tactics of the six fittest morphologically heterogeneous teams in the Zone Attack scenario. T: number of tank robots, S: number of scouts, A: number of artillery.

Iteration	T	S	A	Tactics	Fitness
1	1	4	0	Three-pronged attack	.226
2	1	4	0	Three-pronged attack, synchronized	.230
3	1	4	0	Three-pronged attack, synchronized	.217
4	1	4	0	Three-pronged attack, synchronized	.213
5	1	4	0	Three-pronged attack, synchronized	.216
6	1	4	0	Homogeneous charge	.197

4.2.3 Evaluation with Control Teams

Since only the attacking team was evolved in Zone Attack, the situation confronted by each team during each experiment iteration was invariant. It was therefore possible to use control teams with unchanging behaviors and consistent performance as baselines to assess the evolved teams. Each control team followed one of two homogeneous group behaviors:

1. **Control_Charge:** Each robot takes the shortest path from its starting position to the target zone, moving as quickly as possible.

2. **Control_Flank:** All robots perform a flanking maneuver, hugging the West side of the map and turning in towards the target zone when they come within 30 meters of the map's Northern edge.

For heterogeneous teams the control teams comprised five scouts, while the homogeneous controls used tanks. Each control team and each evolved team was assessed across 1000 rounds of the Zone Attack scenario. Their average fitness values and rates of victories were recorded for comparison, and used to assess whether the tactics evolved by experimentation showed any advantage over the baselines.

For homogeneous experiments the answer was affirmative. Simple charging or flanking tactics resulted in disaster for morphologically homogeneous teams of tanks.

The charging control team won just two rounds out of a thousand, and the flankers won none at all. The teams with evolved behaviors, by contrast, won between 88 and 128 games per thousand (see Figure 18).

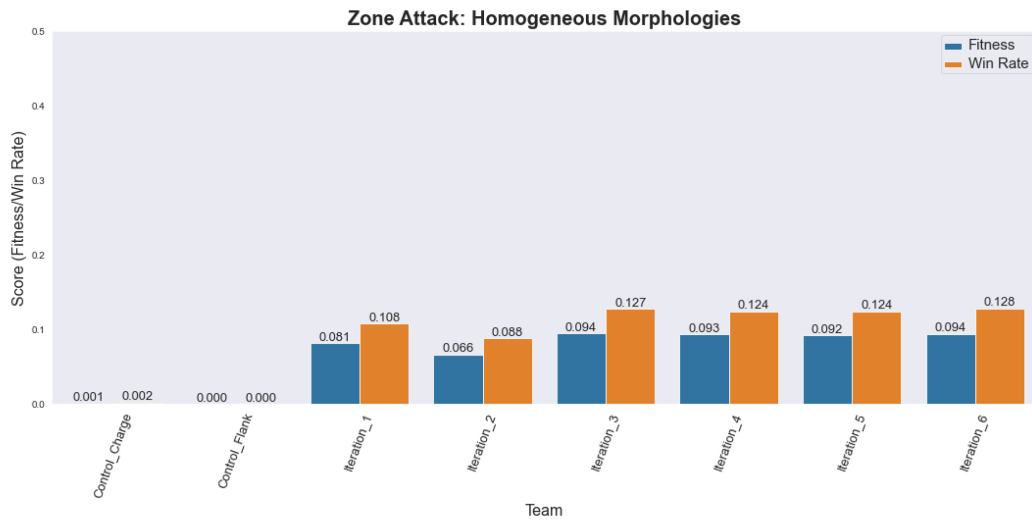

Figure 18: Fitness scores and win rates for morphologically homogeneous teams in Zone Attack.

By contrast, performance of a charging team of scouts was comparable to that of the evolved, morphologically heterogeneous teams (see Figure 19). The result is not surprising given that the genetic algorithm evolved a homogeneous charging strategy on its sixth iteration, and demonstrates the suitability of the scout morphology for this particular task. The flanking control team did not fare as well. This disparity is attributable to two factors: first, flanking units tended to get in each others' way, since they were all following the same path, and second, flankers had to move more slowly to avoid careening into walls when making a turn, making them easier targets.

4.2.4 Summary

Morphologically heterogeneous and homogeneous teams in Zone Attack experiments displayed coordinated heterogeneous behaviors. When allowed to vary mor-

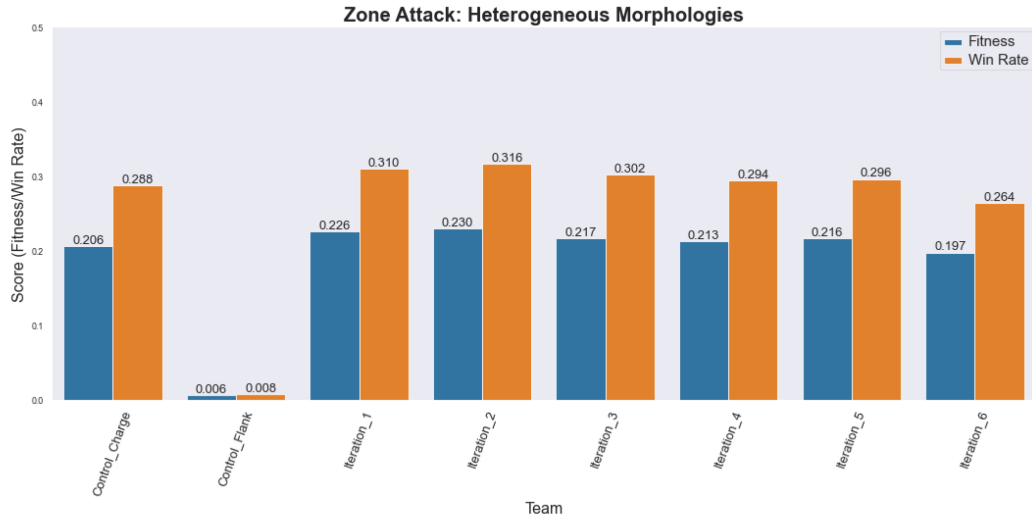

Figure 19: Fitness scores and win rates for morphologically heterogeneous teams in Zone Attack.

phology, the evolutionary algorithm favored morphologically homogeneous teams with large proportions of scouts. These observations indicate that for the Zone Attack scenario, behavioral heterogeneity benefits the attacking team, but morphological heterogeneity does not. The scout appears to be better adapted to the Zone Attack objective than other robot templates, and teams mostly or completely composed of scouts perform comparatively well with homogeneous rush tactics.

4.3 Scenario Two: Zone Defense

Zone Defense is a variation on Zone Attack in which the defending team consists of five tank robots instead of static turrets. The attacking team comprises two scouts. The objective of the attacking team is to get within 30 meters of the defended point at (100,200) before 25 seconds have elapsed, and the objective of the defending team is to prevent the attacking team from achieving its aims. The positions of the defenders are assigned statically as shown in Figure 20, but the attacking robots were placed randomly within 20 meters of the map's Southern edge.

Both the attacking and defending teams were evolved in sequence across eleven iterations. The defenders evolved on odd-numbered iterations, the attackers on even, so that the defenders were evolved a total of six times and the attackers five. At each stage the team evolved by the previous iteration was kept. Consequently each team evolved to counter the tactics of its opponents' latest evolution. Win rates and fitness values were calculated for each evolved team across 1000 scenario repetitions.

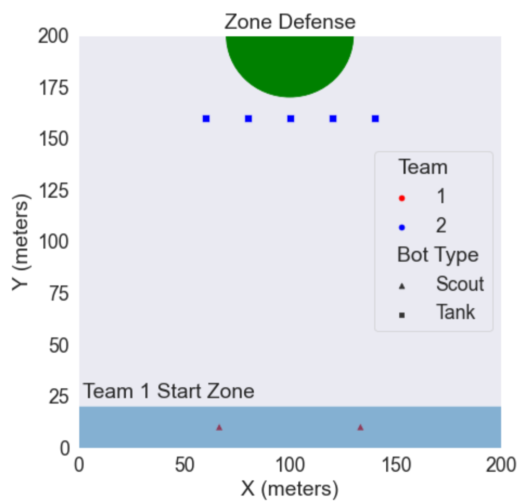

Figure 20: Zone Defense starting positions

4.3.1 Morphologically Homogeneous Teams

In morphologically homogeneous iterations, the defenders' evolved tactics all fit into one of two rough categories. To prevent the two enemy scouts from reaching the target zone, the defenders would physically obstruct the attacker's route with one or more tanks, or else catch the attackers in a crossfire as they approached the zone. Heterogeneous behaviors were in evidence, particularly when defenders set up a crossfire. One such crossfire occurred in iteration 5, the defenders' third evolution. Four of the defending tanks arranged themselves in a roughly triangular shape, with one face almost parallel to the East flank. When an attacking scout approached on

that flank, the two tanks closest to the enemy relayed its position to the two tanks furthest away, allowing all four to fire on the enemy simultaneously (see Figure 21).

Figure 21: Defenders catch a flanking scout in a crossfire.

Defenders performed complex cooperative maneuvers in order to obstruct routes to the target zone. In iteration 11, the defending team evolved an intricate three-tank maneuver to block a Western flanking movement by the attackers (see Figure 22). This maneuver required tank 5 to fire on tank 3, its ally. A robot with no control signal relating to movement will start to back up if struck by a projectile or another robot. Consequently 5's attack on 3 caused the latter to back up until it was parked against the map's Northern edge. Meanwhile tank 4 began a clockwise patrol of the map perimeter. This caused 4 to collide with 3 just outside the radius of the defended zone. When the flanking scouts began their approach, they collided with 3 and 4, and were prevented from reaching the objective before time ran out. None of the available atomic behaviors would have caused 3 to back up and stop where it did, or induced 4 to halt at a particular point on its patrol. The obstructing behavior arose from an interaction which would not have been possible sans any of the three participating agents.

Other heterogeneous behaviors included tasking different defenders to intercept or obstruct attackers on different routes. Since the attacking team only comprised two robots its options were more limited, and the only heterogeneous behavior observed

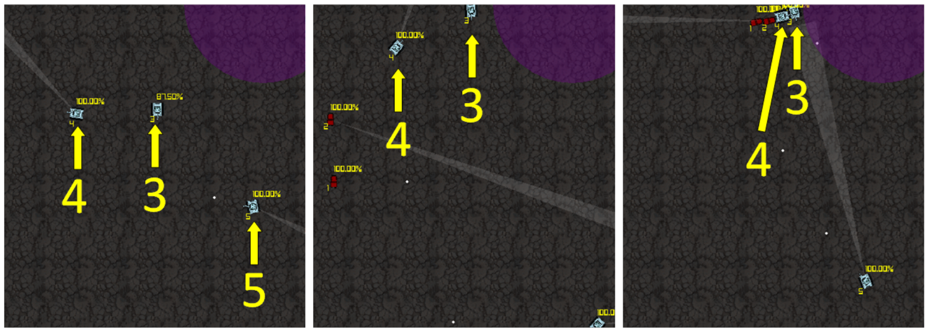

Figure 22: Defenders perform a multi-step maneuver to obstruct the target zone.
Step 1: Robot 5 shoots 3, causing it to begin backing up, while 4 starts driving clockwise around map perimeter
Step 2: Robot 3 backs into the map's Northern edge, blocking 4's path
Step 3: The two attacking scouts collide with 4 and 3 outside of the defended zone

was two-pronged approaches in which attackers adopted different routes to the target zone. All of the defending teams' tactics proved brittle, in that the attackers were able to circumvent them by simply varying the direction of approach. The attacking team responded to the crossfire described above, for example, by changing from an East-side to a West-side flanking maneuver, avoiding the ambush entirely. This caused the East-to-West oscillations apparent in Figure 23, starting at iteration 4.

Morphologically homogeneous teams of defenders performed poorly in most iterations, often failing to evolve a tactic which would stop the attacking scouts in more than a small fraction of test iterations (see Figure 23). The most effective tactics involved heterogeneous behaviors such as staging crossfires or coordinated obstructions, particularly in iterations 5, 9, and 11.

4.3.2 Morphologically Heterogeneous Teams

Morphologically heterogeneous iterations of Zone Defense favored defensive teams which combined a single scout with two to four artillery, while the offense maintained a two-scout morphology for all but one of its iterations. The attacker evolved homo-

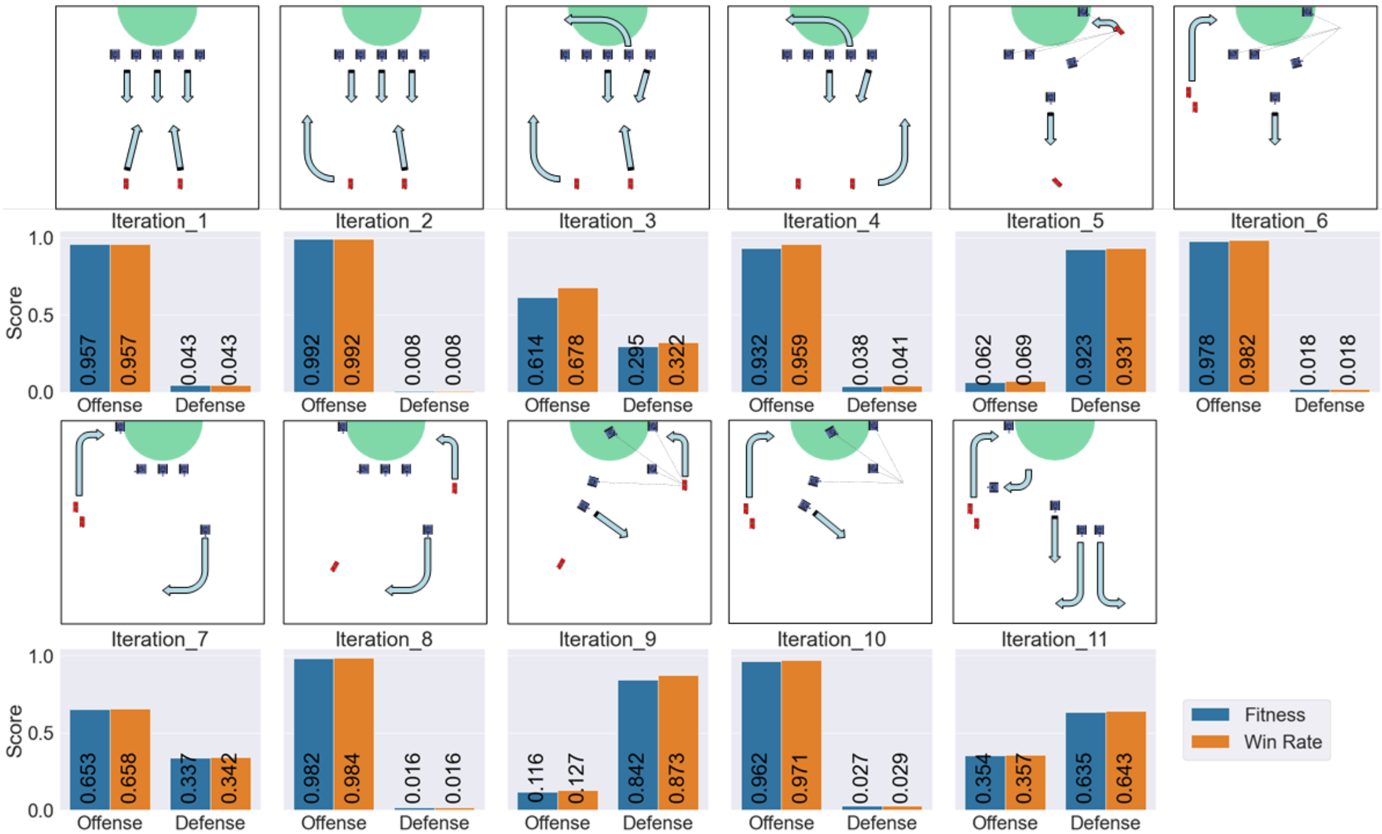

Figure 23: Fitness/win rates for morphologically homogeneous teams, Zone Defense
Iteration_1: Defenders drive forward to obstruct attackers' rush
Iteration_2: Attackers flank West to avoid central defenders
Iteration_3: One defender intercepts West flanker, others target central attacker
Iteration_4: Attacker flanks East to avoid interception
Iteration_5: Defenders catch East flanker in crossfire
Iteration_6: Attackers flank West to avoid crossfire
Iteration_7: One defender obstructs West flank
Iteration_8: Attacker flanks East to avoid obstruction
Iteration_9: Defenders catch East flanker in crossfire
Iteration_10: Attackers flank West to avoid crossfire
Iteration_11: Two defenders obstruct West flankers, two patrol perimeter

geneous charging strategies in iterations 4 and 10, but otherwise stuck to two-pronged attacks. The defenders used their scouts to identify targets for artillery barrage and attempted to obstruct known routes to the target. Often the defense would saturate an area with artillery fire even before an enemy robot was detected. Attacking

units took damage from this blind barrage before entering sensor range, particularly if they attempted a direct approach up the map center. Crossfires and obstructing maneuvers also occurred during evolution (see Figure 24).

Morphologically heterogeneous teams showed a marked preference for mixed team morphologies, particularly combinations of scouts and artillery. They evolved tactics based on heterogeneous behaviors, in which one or more robots obstructed the path of an approaching enemy while other units targeted and destroyed it (see Figure 24, iterations 3 and 9). In iterations 5 and 10, the defenders developed a behavior which transitioned smoothly from blind saturating fire to targeted artillery barrages once adversaries were spotted.

4.3.3 Summary

The fittest teams in Zone Defense employed heterogeneous behaviors, whether those teams were morphologically homogeneous or heterogeneous. Morphologically heterogeneous evolutions exclusively selected teams with mixed morphologies, particularly favoring combinations of scouts with artillery. Consequently it is apparent that both behavioral and morphological heterogeneity is advantageous to defending teams in the Zone Defense scenarios. As the scout is incapable of dealing damage and the artillery of identifying targets, these combinations allow the team to accomplish effects which could not be achieved by either agent type alone. They are therefore an instance of combined arms as described in Army doctrine [1].

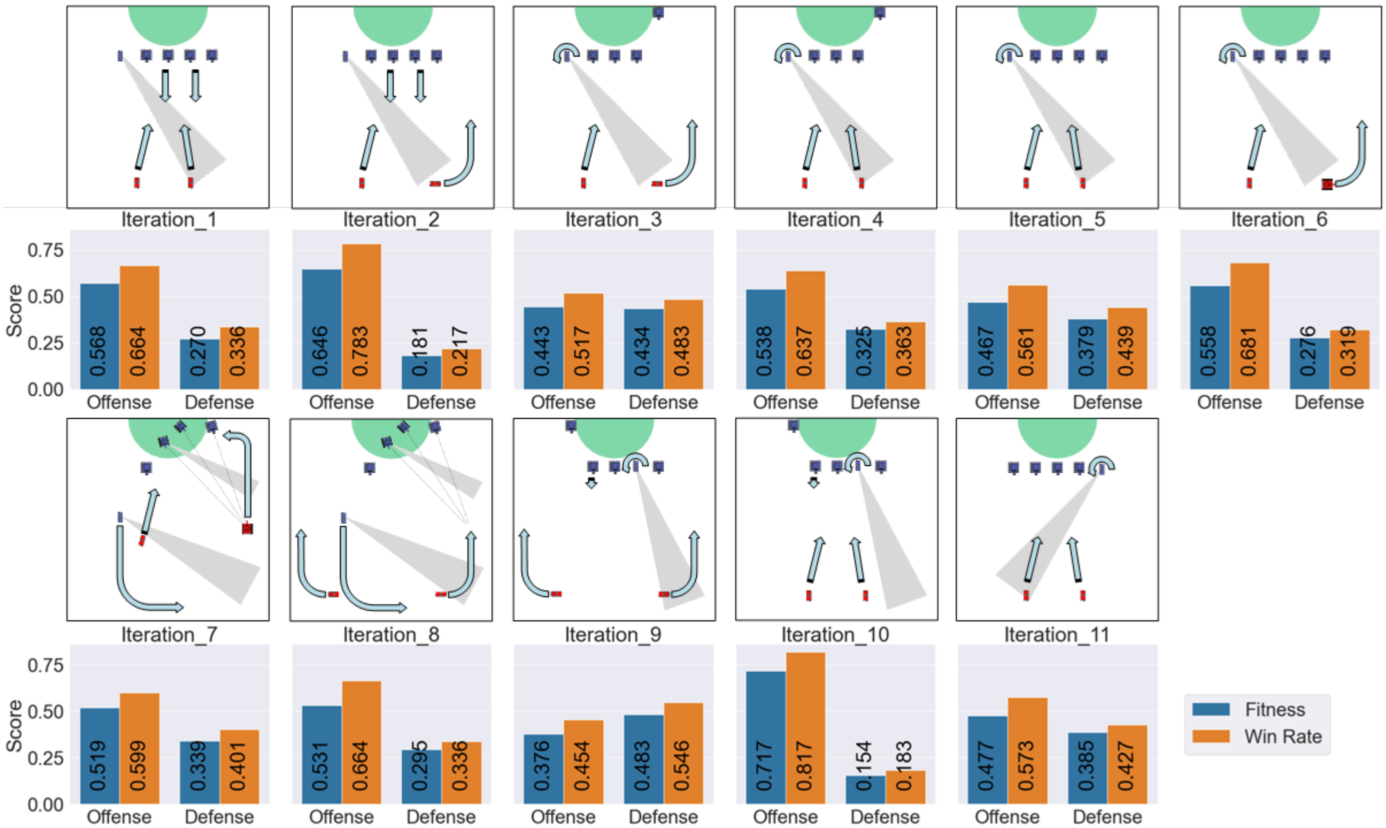

Figure 24: Fitness/win rates for morphologically heterogeneous teams, Zone Defense
Iteration_1: Defending scout identifies targets for artillery barrage
Iteration_2: One attacker charges center, one flanks East
Iteration_3: Scout rotates for better sensor coverage, one artillery obstructs East
Iteration_4: Attackers charge directly
Iteration_5: Defending artillery fire blind until scout identifies targets for barrage
Iteration_6: One attacker charges center, one flanks East
Iteration_7: Defending artillery fires blind in center, then artillery & tanks catch East flanker in crossfire
Iteration_8: Attackers flank West and East in pincer
Iteration_9: One defender obstructs West, scout identifies targets for artillery barrage
Iteration_10: Attackers charge directly
Iteration_11: Defending artillery fire blind until scout identifies targets for barrage

4.4 Scenario Three: Elimination

In Elimination scenarios, the attackers' objective is to destroy a designated robot, hereafter referred to as 'the target', on the defending team, and the defenders' objec-

tive is to keep the target from being destroyed for 25 seconds. The attacking team is provided no indication of the target's location, forcing it to first locate and then destroy the target in each round. The target starts the game at coordinates (100, 160) on the map. During morphologically heterogeneous evolution the defending team may change the target's type just as with any other robot.

Elimination experiments were conducted with small and large teams. In the small team experiments, the defending team was given five robots, including the target, while the attackers had four. In large team experiments, the defending team had eleven robots and the attackers seven. The defenders enjoyed a numerical advantage in both cases. The defenders' starting positions were static, with defenders arranged around the target. The attackers' positions were randomized for each round, each attacker starting at a random position within 20 meters of the map's Southern edge.

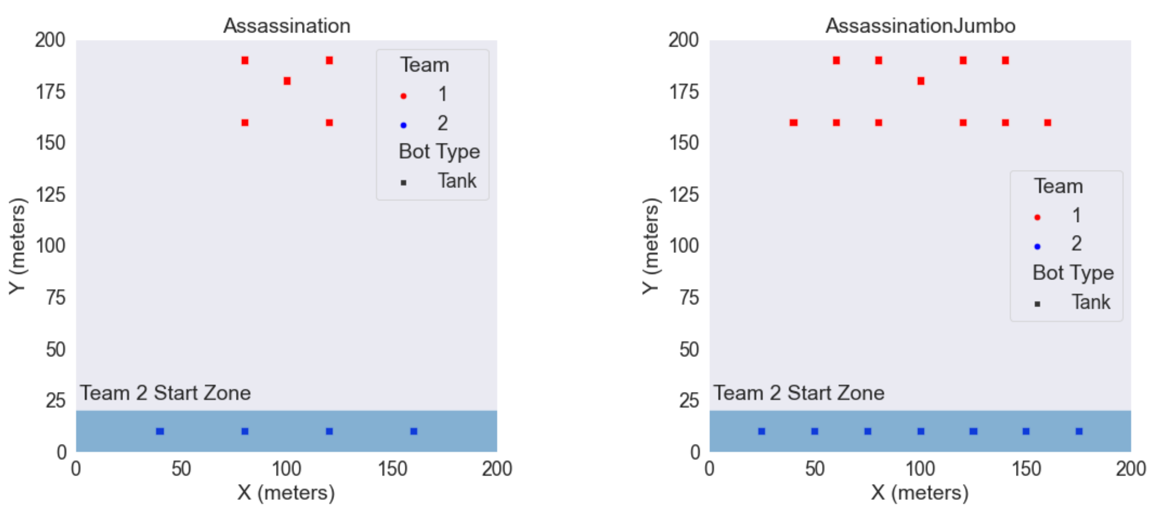

Figure 25: Starting positions for small and large teams in Elimination scenarios

4.4.1 Morphologically Homogeneous Teams

4.4.1.1 Small Teams

Small, morphologically homogeneous offensive teams favored multi-pronged at-

tacks or flanking movements when the target stayed in one place, as in iterations 1, 5, and 6. When the target was mobile the attacking team attempted to intercept it with a flanking movement, as in iterations 9 and 11, or with a crossfire as in iteration 3. Heterogeneous behaviors were readily evident in the multi-pronged attack and crossfire, though in some cases homogeneous flanking movements, even with a single robot, proved sufficient to the task at hand.

The defenders' countered most attacks by placing the target somewhere the attackers weren't. They accomplished this by simply moving the target out of the line of attack, by obstructing the attackers' route of advance, or in one case by having the target move past the advancing attackers and out of range before it could be targeted (see Figure 26, iteration 2). Heterogeneous behaviors appeared in some iterations but was not always helpful. In iteration 4, for example, a robot moves to obstruct the Western approach even though no enemies are taking that route, and is consequently out of position in iteration 3 when the offense flanks East.

4.4.1.2 Large Teams

The offensive team in large, morphologically homogeneous iterations began with a heavy homogeneous flanking movement. While not an example of heterogeneous behavior or of combined arms per se, this movement did place the attacking robots in position to enfilade the defenders, attacking on the long axis as described in USMC doctrine [73] (see Figure 27). Once the flanking attackers reached the Northeast corner, the defenders began to obstruct each others' fields of fire and had difficulty responding with more than two tanks at a time, mitigating their numbers advantage. Heterogeneous tactics included intercepting the target in a crossfire in iteration 3 and multi-pronged attacks in iterations 7 and 9. The latter two cases are doubtful as examples, however, as the West flankers in iteration 7 and the central attackers in

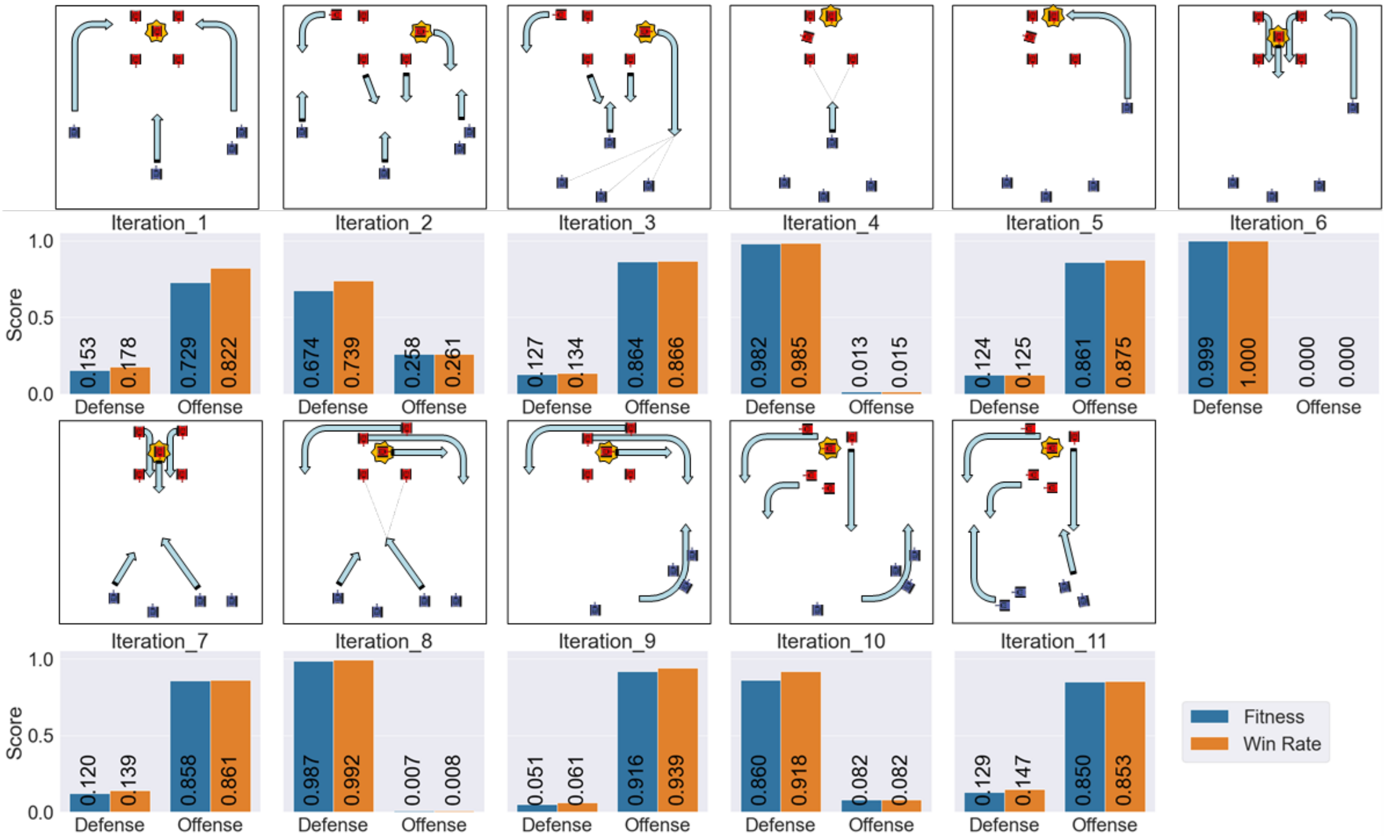

Figure 26: Fitness/win rates for morphologically homogeneous teams, Elimination
Target robot is framed with a yellow star
Iteration_1: Attackers execute a three-pronged assault
Iteration_2: Defenders advance on center & West, target slips past East flankers
Iteration_3: Attackers hold stationary and intercept the target as it moves South
Iteration_4: Defenders destroy attacker in center, target holds stationary
Iteration_5: Single Attacker flanks East, evades central defenders, destroys target
Iteration_6: Target moves to center of map with two tanks escorting
Iteration_7: Attackers concentrate towards map center
Iteration_8: Target flees East, two defenders bar center, one flanks East, one West
Iteration_9: Attackers execute East flank
Iteration_10: Target flees West with three defenders
Iteration_11: Attackers advance in West and center

iteration 9 accomplished the objective with little or no support from the other prongs of the attack. This seems to indicate that a homogeneous strategy may have served just as well.

As with small teams, the typical defensive tactic for a large and morphologically homogeneous team was simply to move the target out of the way. It accomplished this either by moving the target towards the opposite flank of an enemy advance, as in iteration 2, or by moving it to the center, as in iteration 8. Iterations 4 and 8 also developed obstructing behavior, blocking the Western and central approaches with robots in the former case and creating a cluster of robots near the target in the latter. In none of these approaches did the defending robots take on notably heterogeneous roles.

4.4.2 Morphologically Heterogeneous Teams

4.4.2.1 Small Teams

Small, morphologically heterogeneous teams on the offensive side of Elimination scenarios favored combinations of scouts and artillery, with the scouts identifying the target robot so that artillery could destroy it. Iteration 5 evolved a twist on the tactic by stationing two tanks at the map's Southern edge while sending an artillery to intercept the target, which was fleeing South along the West flank. If the target made it past the artillery the two tanks intercepted and destroyed it. Exceptions to the scout and artillery combination occurred in iterations 9 and 11, which eschewed artillery and instead used tanks to pursue or intercept the target as it fled.

The defensive team favored a scout morphology for the target, suggesting that the greater speed and smaller profile of the scout adequately compensated for its reduced energy in this type of scenario. The defenders' tactics were diverse and often relied on combining heterogeneous morphologies and behaviors. Defenders engaged in counter-battery artillery fire in iterations 2 and 8, using a scout to identify targets in the former case and keeping up a blind suppressing fire in the latter. In iteration 2, a tank was sent on a usually fatal charge, during which it swept across the map,

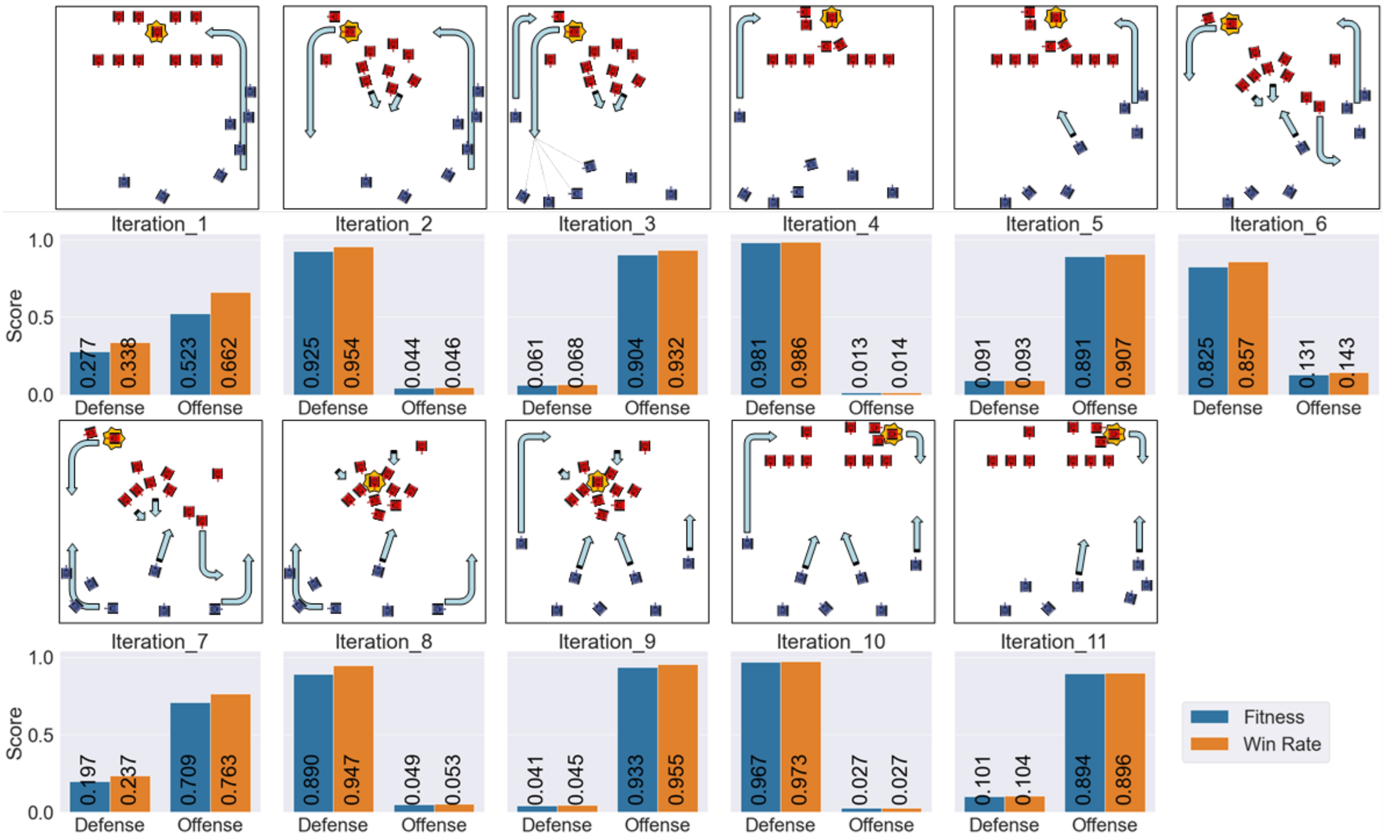

Figure 27: Fitness/win rates for large morphologically homogeneous teams, Elimination
Iteration_1: Attackers flank East
Iteration_2: Target flees West with escorting tank
Iteration_3: Attackers catch fleeing target in crossfire on West flank
Iteration_4: Defenders barricade target on its West and South sides
Iteration_5: Attackers flank East, avoiding barricading defenders
Iteration_6: Target flees West
Iteration_7: Attackers move forward in three prongs, focusing on West Flank
Iteration_8: Defenders cluster in map center
Iteration_9: Attackers move forward in three prongs, focusing on center
Iteration_10: Target flees East with two defending tanks
Iteration_11: Attackers flank East, intercepting target

fired continuously, and drew fire from the attacking artillery. This gave the defending artillery a brief grace period during which it could target the attacking artillery while they focused on the decoy (see Figure 28, iteration 2).

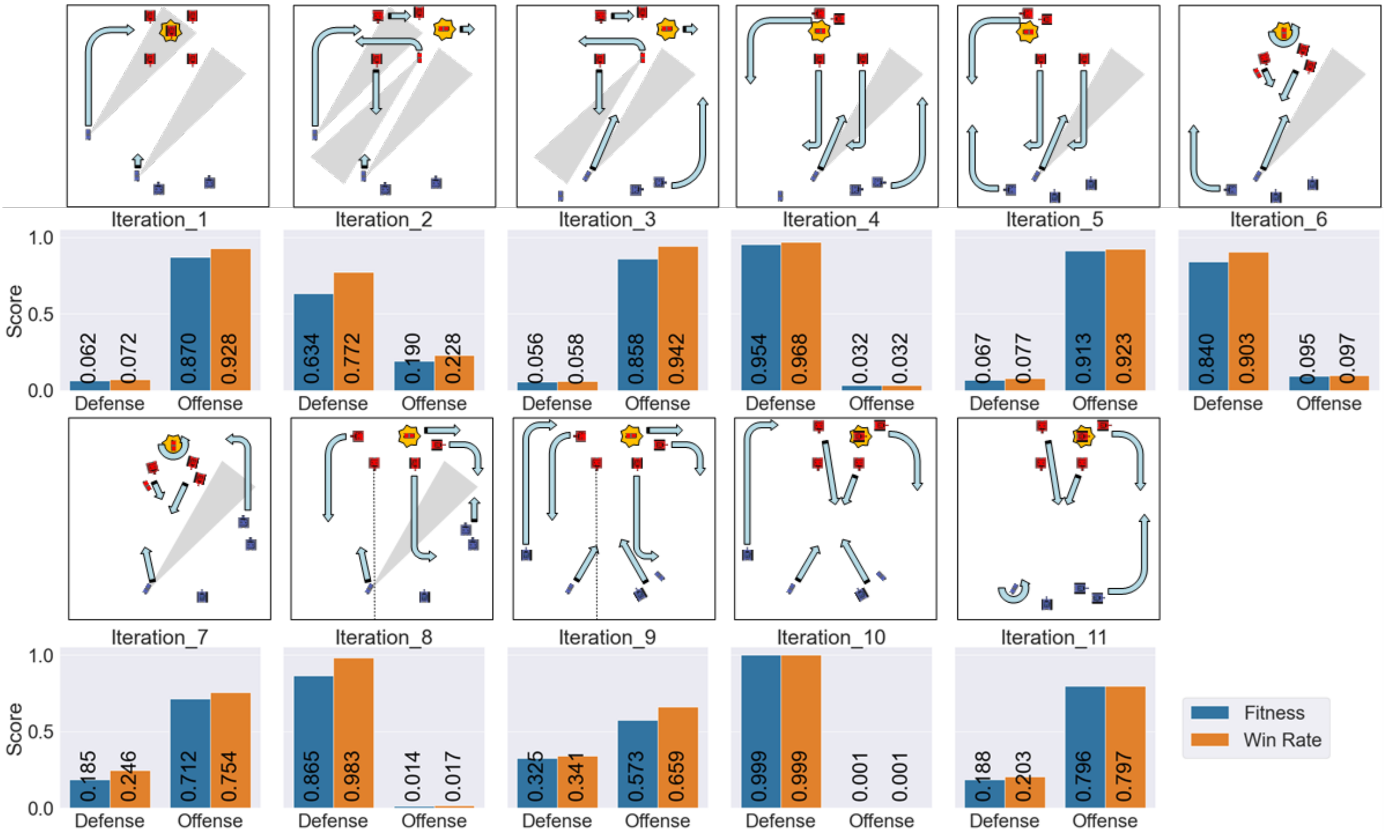

Figure 28: Fitness/win rates for morphologically heterogeneous teams, Elimination
Iteration_1: Attacking scouts identify target for artillery barrage
Iteration_2: Target flees East, defending scout spots targets for artillery, tank charges towards the enemy team
Iteration_3: Attacking scout spots targets as artillery flanks East to intercept target
Iteration_4: Target flees West with two defenders
Iteration_5: Attacking scout spots targets for flanking artillery, stationary tanks
Iteration_6: Target moves in tight circles, other defenders concentrate in center
Iteration_7: Attacking scout spots targets while two artillery flank East, avoiding central defenders
Iteration_8: Two defending tanks engage flanking artillery with pincer
Iteration_9: Attacking tank flanks West around defenders to destroy target
Iteration_10: Target flees East with tank, keeping out of reach of attacking flanker
Iteration_11: Attackers flank East

4.4.2.2 Large Teams

On offense, large morphologically heterogeneous teams again combined scouts

with artillery in iterations 1 and 3. The other four evolutions produced teams that used groups of tanks to intercept the target on its chosen line of retreat, and made little or no use of artillery at all. Another crossfire evolved in iteration 9. This and the collaborations between scout and artillery comprised the only noticeable heterogeneous behaviors of the offensive teams.

The first two evolved defensive tactics were noteworthy. The tactic of blind suppressing artillery fire resurfaced in iteration 2, but with a larger team the defenders were able to dedicate six artillery to the task, creating a devastating barrage. The target juked in tight circles while the defending artillery showered the attackers' side of the map with shells. Even after the offense evolved an effective response in iteration 3, it still lost more than two thirds of its robots' combined energy, on average. Another defensive tactic involved sending tanks to the center of the map to draw artillery fire. The attackers were sending their scouts to the map center, and with the tanks in their near vicinity, they tended to fall prey to attacks from their own artillery. Meanwhile the target hid in the Northwest corner. once the attacking scouts were destroyed, the attackers had no way to detect the target in its hiding place (see Figure 29). Aside from these two instances the defensive teams evolved relatively simple behaviors which sent the target fleeing in one direction and occasionally attempted to entangle pursuers with other defenders.

4.4.3 Summary

Both offensive and defensive teams in the morphologically heterogeneous Elimination experiments displayed heterogeneous behaviors and complementary morphologies, and furthermore provided examples of combined arms behavior. The attackers in iteration 5 and the defenders in iteration 2 of the morphologically heterogeneous small-team experiments gave particularly notable examples of scenarios in which robots of

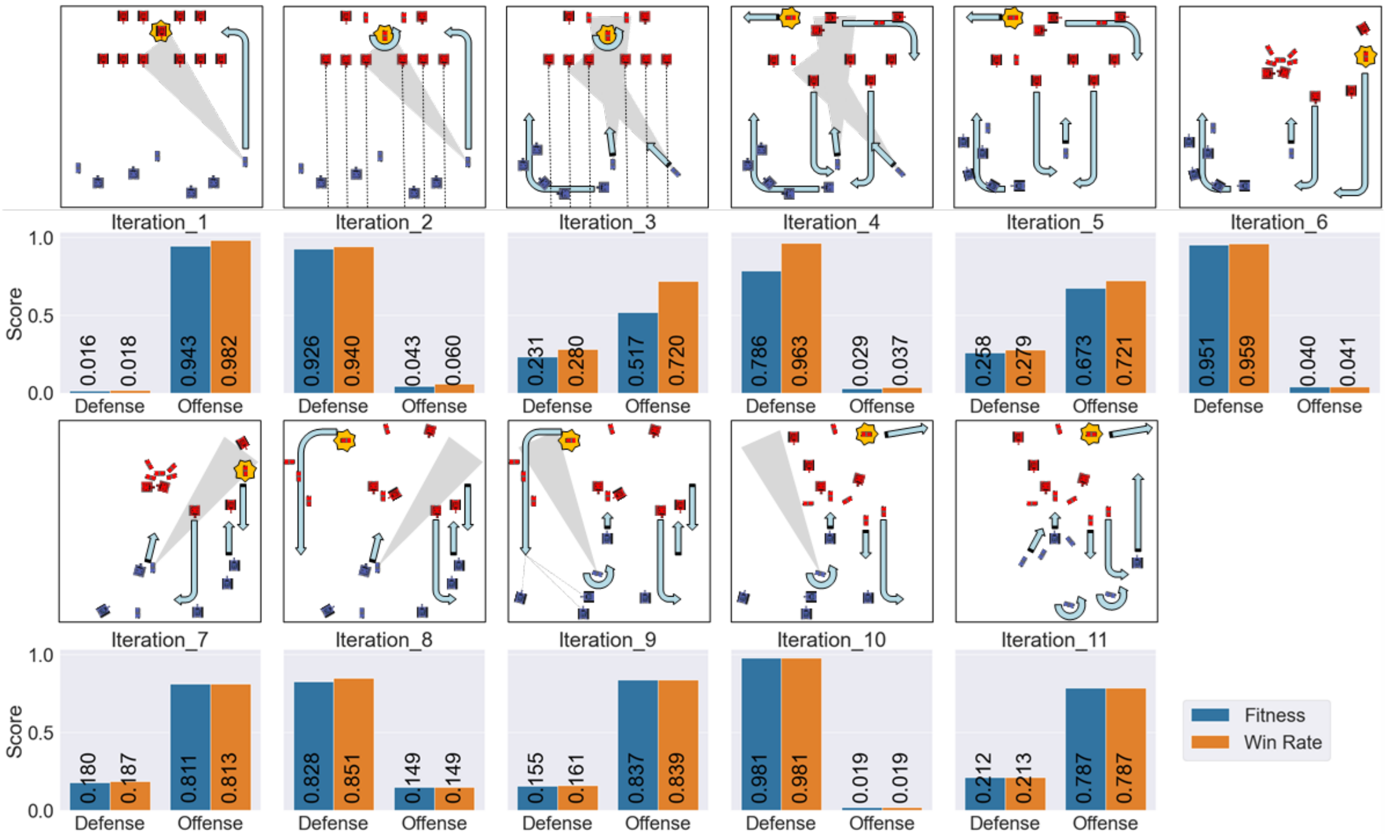

Figure 29: Fitness/win rates for large morphologically heterogeneous teams, Elimination

Iteration_1: Attacking scout identifies target for artillery barrage

Iteration_2: Defenders saturate South map edge with artillery fire, target moves in tight, evasive circles

Iteration_3: Attackers flank West to avoid artillery fire

Iteration_4: Two defending tanks decoy artillery & destroy scouts, target flees West

Iteration_5: Attackers change all artillery to tanks, execute heavy West flank

Iteration_6: Target flees East

Iteration_7: Attackers flank East, send artillery and scout to center

Iteration_8: Target flees West

Iteration_9: Attackers catch target in crossfire on West flank

Iteration_10: Target hides in Northeast corner

Iteration_11: Attackers send tank to Northeast corner

all three morphologies were combined to execute maneuvers which no single type could have performed alone. Iteration 5 saw the attackers employing tanks as insur-

ance for an initial attack with scout and artillery, so that in escaping artillery the target ran into the attacking tanks. In iteration 2 the attacking artillery were forced to deal with the tank's charge, and in so doing exposed themselves to destruction by defending artillery. This tactic exemplifies the principle laid out in USMC doctrine [83]: "The use of combined arms places the enemy in a dilemma. Any action the enemy takes to avoid one combat arm makes him more vulnerable to another."

4.5 Scenario Four: Last Team Standing

The objective in Last Team Standing scenarios is to destroy all robots on the opposing team within 83 seconds[1]. Since neither team can be designated 'offense' or 'defense', they are hereafter referred to as the 'Northern' and 'Southern' teams according to their starting zones on the map (see Figure 30). Each team comprised five robots[2] and the starting positions of both teams were randomized, each robot starting within 20 meters of its assigned map edge.

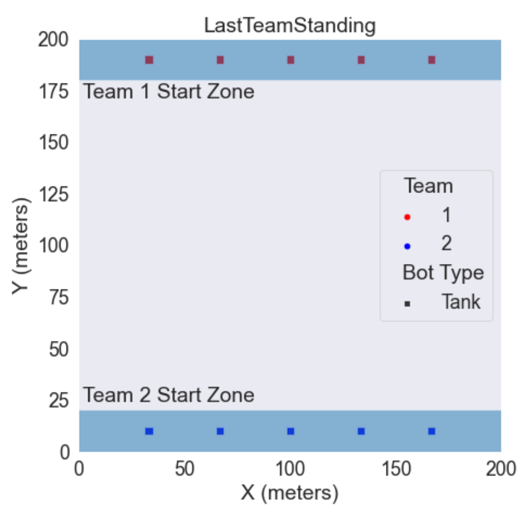

Figure 30: Last Team Standing starting positions

[1]5000 ticks of the physics engine

[2]Experiments with larger teams resulted in collisions between allies and prevented robots from executing their evolved behaviors, so that large team iterations of Last Team Standing turned into amorphous brawls in the restricted space of the battlefield.

4.5.1 Morphologically Homogeneous Teams

Most evolutions of the Southern team produced three-pronged attacks, sending one or two tanks each on West, East, and central lines of approach. In the first iteration this maneuver had the advantage of catching the Northerners in enfilade from two directions at once, though that advantage disappeared when the Northern team evolved more mobile strategies. From there, neither teams' tactics showed very much variation. Teams' adjusted by modifying the number of tanks assigned to each of four general roles: West flanker, East flanker, central charger, or stationary interceptor. The first three roles simply refer to the route the tank took to reach the enemies' side of the map. The last denotes tanks that stuck close to their starting location and typically intercepted enemy flankers as they advanced. No iteration produced a wholly homogeneous set of behaviors for either team. In all cases, tanks were assigned to at least two and usually three of the listed roles.

Tied matches were fairly common. Often one or more tanks from each team would survive until the match end, resulting in no victory for either team. Consequently the win rates shown in Figure 31 do not add to one.

4.5.2 Morphologically Heterogeneous Teams

Morphologically heterogeneous evolution produced several teams combining scouts and artillery, with some interesting variations. Twice the Northern team sent artillery to the center of the map. This placement shortened the distance shells had to travel to hit most targets and allowed the artillery to reach every corner of the map, but left the robots more exposed than they might have been on the back rank. In iteration 5 the Southern team did away with scouts and used tanks to identify targets for artillery, sacrificing sensor range for a combat-capable platform. Morphologically homogeneous teams comprising exclusively tanks evolved twice, once for the South in

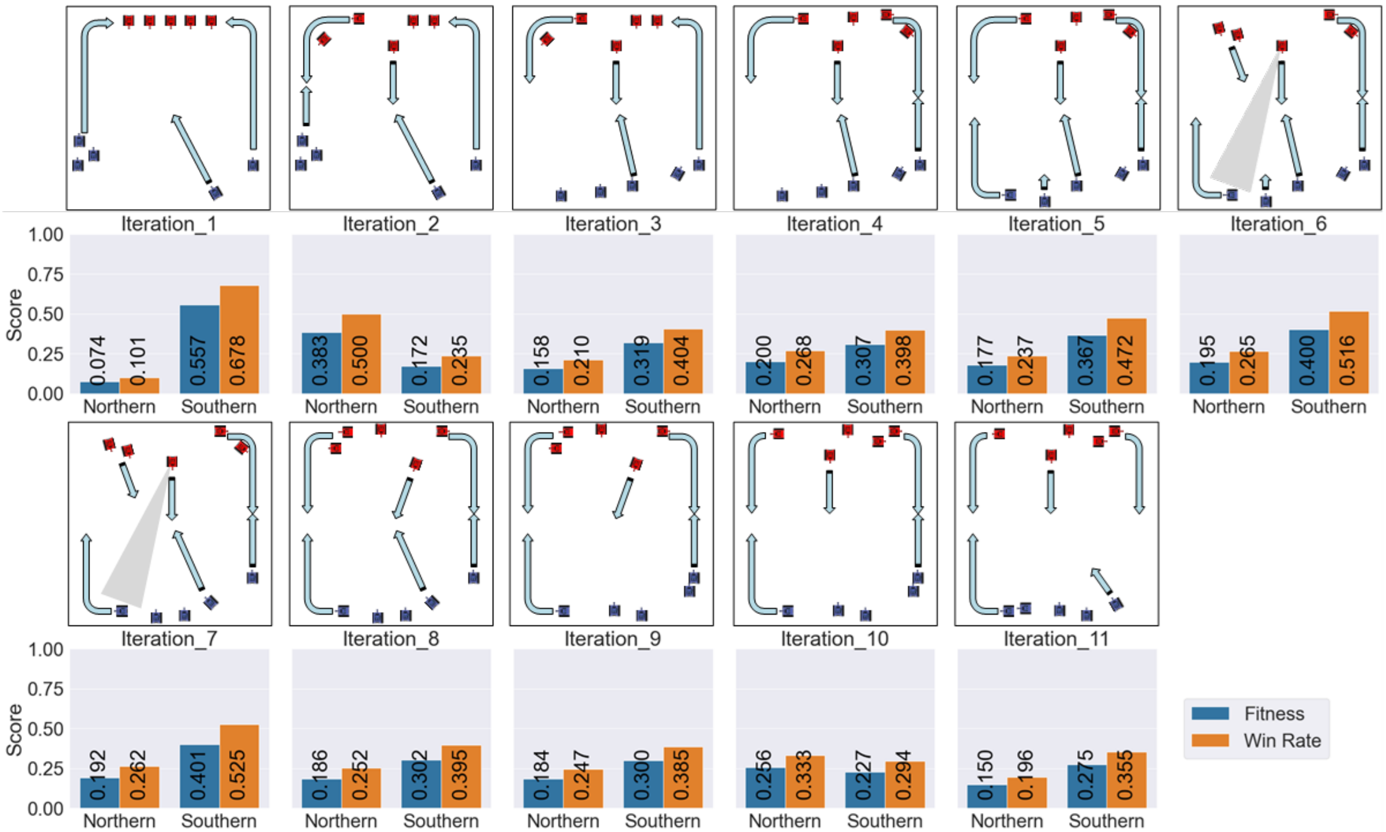

Figure 31: Fitness/win rates for morphologically homogeneous teams, Last Team Standing
Iteration_1: South executes three-pronged attack, favoring West flank
Iteration_2: North counters on West and center lines, leaves two tanks stationary
Iteration_3: South adds one tank to East flank, stops flanking West
Iteration_4: North executes three-pronged attack
Iteration_5: South executes three-pronged attack, deadlock
Iteration_6: North sends three tanks to center, two down East flank
Iteration_7: South switches one tank from East flank to stationary
Iteration_8: North returns to three-pronged attack
Iteration_9: South flanks East and West, keeps two tanks stationary
Iteration_10: North moves one tank from West to East flank
Iteration_11: South flanks West, sends one tank to center

iteration 2, and for the North in iteration 7. In both cases the tanks proved effective at using multi-pronged advances to engage vulnerable artillery or scout robots. All evolutions displayed behavioral heterogeneity, and all except the tank-only iterations

above displayed morphological heterogeneity.

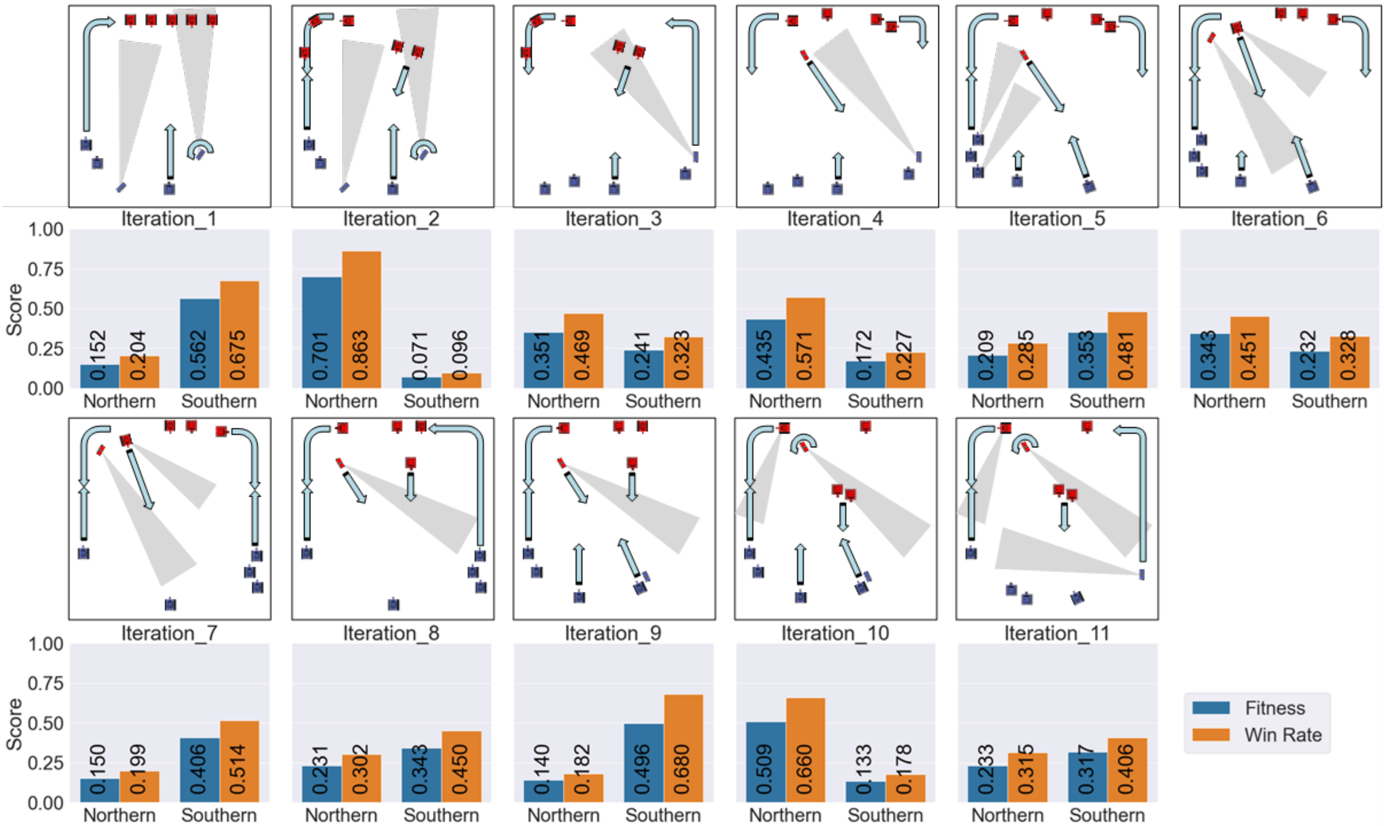

Figure 32: Fitness/win rates for morphologically heterogeneous teams, Last Team Standing

Iteration_1: South flanks West with artillery & tank, uses scouts to spot targets for central and flanking artillery

Iteration_2: North flanks West with three tanks, sends two tanks to center

Iteration_3: South uses patrolling scout to spot targets for four artillery

Iteration_4: North executes West/East pincer

Iteration_5: South flanks West with three tanks, tanks spot targets for artillery

Iteration_6: North flanks West with scout and tank, scout spots targets for artillery

Iteration_7: South executes pincer, favoring East flank & selecting an all-tank team

Iteration_8: North sends artillery and spotting scout to center, one artillery West

Iteration_9: South flanks West with two tanks, moves two tanks to center

Iteration_10: North moves two artillery to center, keeps spotting scout at rear

Iteration_11: South sends scout East to spot targets for two artillery at rear

4.5.3 Summary

Heterogeneous behaviors emerged in both morphologically homogeneous and heterogeneous iterations, with multi-pronged attacks in common use. Heterogeneous morphologies were generally preferred in Section 4.5.2, but both teams produced a morphologically homogeneous team during one of their iterations. Elements of combined arms tactics were evidenced whenever scouts or tanks were used to identify targets for artillery. No tactics were observed which had not occurred in previous experiments.

4.6 Fitness Comparisons

Another set of experiments was carried out independently of those described in the last four sections, to compare the fitness scores of morphologically homogeneous teams with those of morphologically heterogeneous teams. Subject teams of both morphological types were evolved for each scenario. The subject teams were as follows:

1. Zone Attack: Attacking team
2. Zone Defense: Defending team
3. Elimination (both team sizes): Attacking team
4. Last Team Standing: Southern team

The subject team was evolved after each iteration, without morphological mutation. Consequently both evolved teams competed against the same attacking team in every round. Fitness scores were assessed over 1000 game rounds following each evolution. Single-tailed Student's T-Tests with $\alpha = 0.05$ were performed on the results to verify statistical significance. The results show that teams evolved with heterogeneous morphologies enjoyed a statistically significant ($p <= 0.0001$) advantage in

fitness over teams evolved with homogeneous morphologies in all scenarios except for Last Team Standing (see Table 4).

Table 4: Parameters and p-values for Student's T-Tests of Fitness scores for morphologically homogeneous and heterogeneous teams. μ: Mean Fitness of 6 evolved teams across 1000 games each; CI: Confidence Interval; p: p-value generated by Student's T-Test

Morphology	**Homogeneous**		**Heterogeneous**		
Scenario	μ	**CI** ($\alpha = 0.05$)	μ	**CI** ($\alpha = 0.05$)	p
Zone Attack	.0747	.0681, .0814	.2193	.2088, .2297	$< .0001$
Zone Defense	.1580	.1488, .1672	.3411	.3291, .3531	$< .0001$
Elimination	.8258	.8162, .8354	.9015	.8940, .9090	$< .0001$
Elimination (Large Teams)	.8430	.8338, .8522	.8523	.8433, .8613	.0001
Last Team Standing	.3748	.3626, .3871	.3604	.3483, .3726	.8702

4.6.1 Zone Attack

Morphologically heterogeneous teams showed a clear advantage in fitness for Zone Attack scenarios. However, it must be remembered that evolution produced morphologically homogeneous teams of scouts in nearly all cases, as shown in Section 4.2. The score disparity indicates the advantage of scouts over tanks, rather than of morphological heterogeneity.

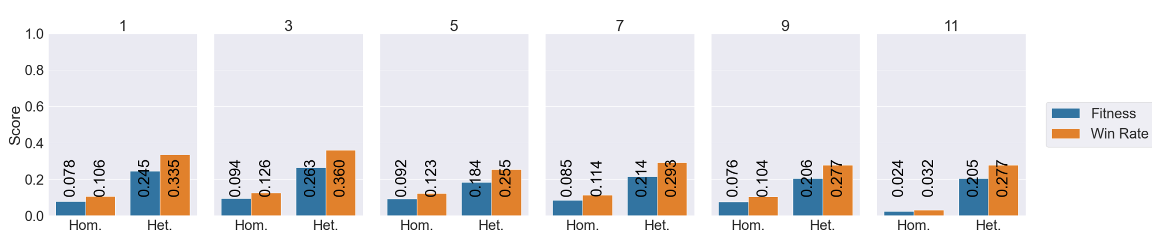

Figure 33: Fitness and Win Rate scores for morphologically heterogeneous and homogeneous defending teams when evolved against identical attacking teams in the Zone Attack scenario.

4.6.2 Zone Defense

Tests for Zone Defense showed a significant increase in average fitness for morphologically heterogeneous teams. The morphologically homogeneous team performed better during a single iteration (see Figure 34, iteration 5). The attacking scouts in that test case used a simple charging strategy that took them straight down the map's center. The morphologically homogeneous team's five tanks formed a stationary wall in front of the target zone, and were often able to destroy both scouts as they advanced. The morphologically heterogeneous team formed the same wall with artillery and a single scout, and the attacking robots were able to slip past the slower-firing artillery more often than the tanks. In all iterations in which the attacking team adopted a flanking or multi-pronged approach, however, the morphologically heterogeneous team excelled the morphologically homogeneous.

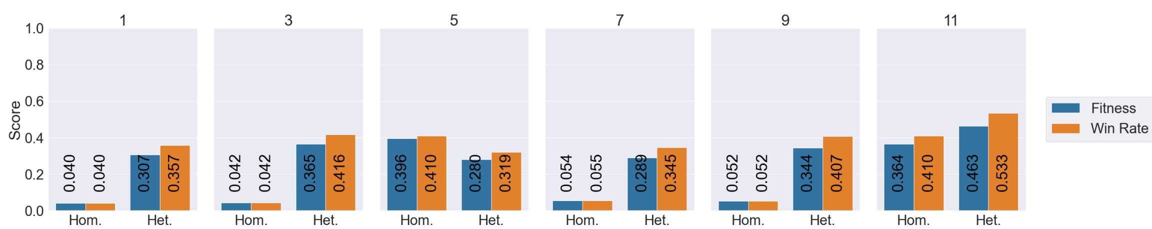

Figure 34: Fitness and Win Rate scores for morphologically heterogeneous and homogeneous defending teams when evolved against identical attacking teams in the Zone Defense scenario.

4.6.3 Elimination

Morphologically heterogeneous teams performed better in Elimination scenarios with both small and large teams, but the difference in fitness was smaller with large teams. The morphologically homogeneous teams performed better in three rounds. In two of those cases the morphologically heterogeneous team evolved a homogeneous morphology. In the third case, which took place with large teams, the homogeneous

team employed a more aggressive strategy, with tanks first dispersing and then converging on the location of the target. This worked better than the heterogeneous team's approach of ambushing the target when it reached the Southern map edge.

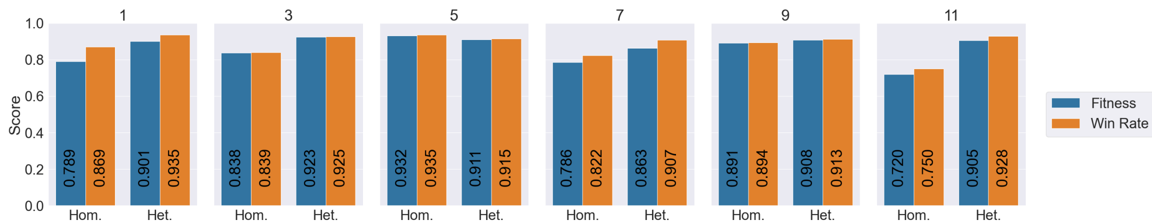

Figure 35: Fitness and Win Rate scores for morphologically heterogeneous and homogeneous defending teams when evolved against identical defending teams in the Elimination scenario with small teams.

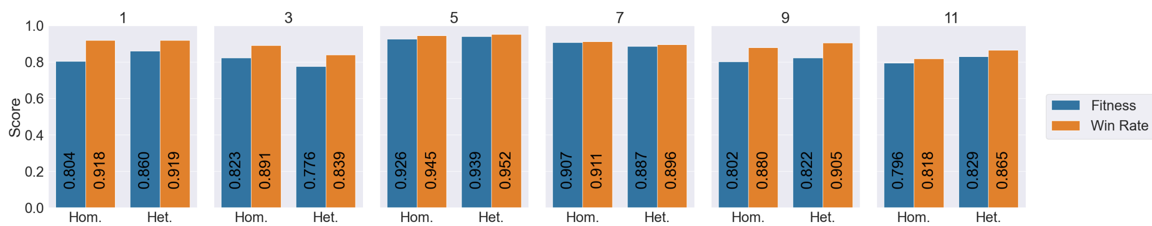

Figure 36: Fitness and Win Rate scores for morphologically heterogeneous and homogeneous defending teams when evolved against identical defending teams in the Elimination scenario with large teams.

4.6.4 Last Team Standing

The fitness comparison for Large Team Standing is unique in that it shows a slight advantage for morphologically homogeneous teams. When the Student's T-Test tail is reversed to test whether morphologically heterogeneous values are significantly less than the morphologically homogeneous scores, the test returns $p = .1298$, not low enough to guarantee statistical significance. Nevertheless, the test indicates that in Last Team Standing, morphologically homogeneous evolution produces teams roughly as effective if not more effective than morphologically heterogeneous teams.

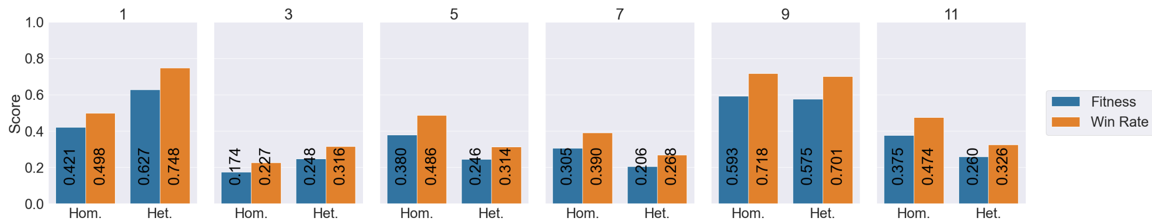

Figure 37: Fitness and Win Rate scores for morphologically heterogeneous and homogeneous defending teams when evolved against identical Northern teams in the Last Team Standing scenario.

4.6.5 Summary

Morphologically heterogeneous evolution conducts mutation and crossover with a set of behaviors and a set of robot templates, while morphologically homogeneous evolution only modifies behaviors. There is no difference in the behavior set used by either evolution type. Therefore any team produced by morphologically homogeneous evolution can also be produced by morphologically heterogeneous evolution, though the latter method may take longer to produce a given configuration because the domain is larger.

The Last Team Standing results, and the iterations in other scenarios during which morphologically homogeneous teams attained superior fitness scores, suggest that the constraints of morphologically homogeneous teams may allow them to adapt to a given scenario in fewer evolutionary iterations. The result brings to mind Patton's observation that "a good plan violently executed now is better than a perfect plan next week" [84]. If a similar method were applied to combined arms tactics, it might be desirable in urgent circumstances to limit variables like morphology in order to obtain more rapid solutions.

4.7 Summary

This chapter discussed results obtained from four experimental scenarios. It out-

lined emergent tactics observed in evolved teams, including flanking behaviors and morphology-based division of labor, and provided a detailed account of experimental runs for each scenario and team composition, demonstrating the emergence of particular tactics suited to particular objectives. The problems of oscillation in homogeneous Elimination engagements and deadlock in various Last Team Standing scenarios were noted, and potential correctives will be suggested in Chapter V.

V. Conclusions

This final chapter revisits each of the research questions in turn and draws answers from the data in Chapter IV. It assesses the original hypothesis of this research and proposes lines of effort for future work.

5.1 Behavioral Heterogeneity

Does behavioral heterogeneity improve or impair the performance of multi-agent systems in a combined arms scenario? The value of behavioral heterogeneity for a given team seems to depend on the tactical situation. General Mattis' oft-repeated observation that "the enemy gets a vote" [85] remains in force. Homogeneous flanking maneuvers in which every member of the team follows the same set route proved effective in several Elimination scenarios, either for applying maximum force to a single enemy flank or for intercepting and swamping a fleeing target. They also proved brittle, however, and led to evolutionary oscillation as the defending team learned to move away from the flanking forces.

Multi-pronged attacks were a frequently recurring example of heterogeneous tactics in homogeneous teams of agents. They proved effective in certain scenarios but sometimes led to deadlock with teams that adopted the same strategy (see Section 4.5). Decoy tactics in which one group of robots distracted defenders while a second group attacked the target were successful in Zone Attack and Elimination scenarios, but defenders in the latter group evolved to ignore the decoys. In brief, no behavior, whether homogeneous or heterogeneous, was discovered for which there was not some counter-tactic.

In light of the above, it would seem that behavioral heterogeneity may or may not improve a team's ability to achieve certain objectives in a certain scenario. The ability

to alter behaviors, changing tactics from homogeneous to heterogeneous as required by circumstance, seemed to be the key requirement. Thus while it is not proven that a behavioral heterogeneous team is more competent by virtue of simple heterogeneity, the ability to grow and evolve into heterogeneous roles allows multi-agent teams to accomplish otherwise impossible objectives.

5.2 Morphological Heterogeneity

Does morphological heterogeneity improve or impair the performance of multi-agent systems in a combined arms scenario? The compositions of teams evolved in these experiments suggest that the value of morphological heterogeneity, like behavioral heterogeneity, seems to rest on the nature of the situation at hand. The fittest teams produced in some iterations of morphologically heterogeneous evolution comprised robots of a single type. Thus morphologically homogeneous teams may be more effective in certain scenarios.

Nevertheless morphologically heterogeneous evolution produced significantly fitter teams in four of the five tested scenarios (see Table 4). Teams evolved diverse morphologies to create effective tactics, like the artillery barrages and decoy behaviors in Section 4.4, which would not have been possible without combining different agent types. The mixing of agent morphologies therefore provides teams with a wider range of tactical options and allows them to develop more specialized and effective configurations.

5.3 Emergence of Combined Arms Tactics

Given a set of morphologically distinct units, can a multi-agent system of software agents exhibit synergistic combined arms behavior without explicit central direction? The answer is yes. Combined arms behavior developed on multiple occasions during

the experiments in Chapter IV. Scouts and tanks employed their sensors to notify artillery of prospective targets. Tanks organized to finish off enemies weakened by artillery fire. Units drew fire away from vulnerable artillery while said artillery laid down a withering barrage from long range. These are all instances in which the simultaneous application of different unit types produced an effect greater than could be expected from their separate and sequential use - neatly fitting the definition of combined arms. These results suggest that multi-agent simulations, perhaps with a more sophisticated and true-to-life simulator, could inform the development of novel approaches to the art of combined arms warfare.

5.4 Future Work

The scenarios used in these experiments gave each team a single objective, such as 'destroy the enemy team' or 'reach the objective point'. Multi-objective scenarios might yield more interesting tactics. Teams could be required to protect an allied target while eliminating an enemy's, for example. One might expect teams to evolve subgroups assigned to different tasks, a set of tanks assigned as bodyguards, for example, while another group hunts the enemy. Such scenarios would bear a certain resemblance to actual military engagements, in which each supporting group of units may have its own set of objectives which contribute to an overall goal.

Multiple tiered objectives could also serve to expand these experiments' scope. The scenarios in this research are 'tactical' in the military sense, each scenario focused on one particular team of units in the performance of a single mission. Military doctrine describes a 'strategic' level of warfare focused on policy objectives rather than individual missions, and an 'operational' level connecting tactical missions to strategic objectives [2]. Simulations could be developed for these two higher levels of warfare and testing the ability of multi-agent systems to develop effective combined

arms behaviors at all levels warfare.

The behavior architecture and evolutionary method used in this research was very basic, involving as it did a composite of four atomic behaviors with their weights, from which the evolutionary algorithm produced novel configurations. An alternative approach could be to associate particular conditions with particular atomic behaviors and use a Learning Classifier System (LCS) for evolution. In such a scheme an Artillery might have a 'CrossMap' behavior associated with a condition, 'if not in range of any adversary'. It would then stop when an adversary was spotted and fire from range. Robots could modify their actions based on such variables the number of spotted enemies, the proximity of allies, or their own remaining energy. Such a set of rules, combined with a larger selection of atomic behaviors, could allow for the development of more intricate team tactics, adaptable to the state of the battlefield.

A future researcher could also explore the effect of adding another factor - terrain - to these simulations. The introduction of obstructing terrain features could greatly modify the calculus of battle and might help eliminate some of the observed problems with deadlock and oscillation. A bottleneck terrain feature would impair an attacking team trying to get at a target robot, but could also prevent the target from simply running in the opposite direction if an attacker made it through. Teams could learn to take shelter from fire behind these obstacles, or to use them for concealment from sensors. This fairly minor modification to the simulator would provide for a wide range of new tactics and might provide the impetus for novel team behaviors.

A final suggestion for future efforts is to test these principles and behaviors in other simulators. Platforms like StarCraft Multi-Agent Challenge (SMAC) provide a wide range of agent types which could provide excellent test cases for both behavioral and morphological heterogeneity [65]. Many such games and simulators incorporate additional domains such as air or naval forces, as well as robust communities of

human players and developers. It would be worthwhile to explore the practicability of developing combined arms tactics within these other platforms.

5.5 Concluding Notes

The specter of the terminator looms large whenever Artificial Intelligence (AI) is discussed in conjunction with military applications. The research conducted here is very far from any practical tool of warfare, representing a simplistic simulation in a simplistic (and flat) world of what is in reality a very complex business. The value of exercises like this one is not that it represents another step towards some kind of automated strategic machine. Rather it is the exploration of a new methodology for informing commanders. When teams in Section 4.4 learned to ignore parts of the adversary team it was noted that such behavior recalled the famous adage that the acme of war is to win without fighting [86]. It is to be hoped that such experiments may help future military commanders find new ways to win conflicts and minimize casualties.

Appendix A. List of Atomic Behaviors

This appendix provides a complete list of atomic behaviors used during evolution, with a brief description of each. Note that for all behaviors which involve firing a weapon, the robot will perform a raycast to see if it is pointing its weapon at an ally, and will only fire if it is not. Friendly fire still occurs as robots move in and out of each others' fields of fire, but this failsafe prevents the worst of it.

1. **Ballerina:** Rotate robot in place, rotate turret, fire weapon, and rotate sensor.

2. **ChargeClosest:** Charge the closest detected enemy robot.

3. **ChargePosition:** If there is a target zone, go as fast as possible on the shortest route to reach that zone.

4. **ChargeRobot:** If there is a target robot and its position is known, accelerate towards that robot.

5. **Concentrate:** Move towards the team's center of mass.

6. **CrossMap:** Move towards the most distant corner of the map.

7. **Disperse:** Move away from the team's center of mass.

8. **DistressCall:** Send a message indicating this robot is in trouble.

9. **Fire:** Fire weapon with given magnitude. The magnitude governs different parameters for different munitions. For Artillery, the magnitude represents the range at which the fired shell is to explode.

10. **Flee:** Move away from the nearest enemy.

11. **Patrol:** Travel around the map perimeter in a clockwise direction.

12. **RandomPatrol:** Travel around the map perimeter in a random direction. Randomly adjust direction and standoff distance when struck by a munition.

13. **RelayTargets:** Notify allies of any enemies this robot has detected.

14. **ReversePatrol:** Travel around the map perimeter in a counter-clockwise direction.

15. **SensorTest:** Rotate sensor in a 360-degree arc.

16. **SittingDuck:** Hold still and take no action.

17. **TargetClosest:** Fire on the closest detected enemy.

18. **TargetRobot:** If there is a target robot and its position is known, fire along heading from this robot to target.

19. **TargetWeakest:** Fire on the detected enemy with the least remaining energy.

20. **Wander_v0** Move in pseudo-random manner, randomly selecting a new heading and velocity after a given number of turns.

21. **WanderFire:** Wander around the map firing blind.

Bibliography

1. United States Army Headquarters. ADP 3-0: Operations. `https://armypubs.army.mil/ProductMaps/PubForm/ADP.aspx`, 2019. [Accessed: 2022.02.21].

2. United States Marine Corps Headquarters. Mcdp 1: Warfighting. `https://www.marines.mil/portals/1/Publications/MCDP%201%20Warfighting%20GN.pdf`, 1997. [Accessed: 2022.02.21].

3. David W. King. *Emergent Behavior Development and Control in Autonomous Swarms*. PhD thesis, 2018.

4. John Holland. *Emergence: From Chaos to Order*. Perseus Books, Reading, MA, 1998.

5. Melanie Mitchell. Complex systems : Network thinking. 170:1194–1212, 2006.

6. Frans P.B. Osinga. *Science, Strategy and War: The strategic theory of John Boyd*. Routledge, New York, 2007.

7. Daniela Kengyel, Heiko Hamann, Payam Zahadat, Gerald Radspieler, Franz Wotawa, and Thomas Schmickl. Potential of heterogeneity in collective behaviors: A case study on heterogeneous swarms. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 9387:201–217, 2015.

8. The Ellis Group. 21st Century Combined Arms. *The Marine Corps Gazette*, dec 2016.

9. James E. Moschgat. *The Composite Wing: Back to the Future!* PhD thesis, Air University, 1992.

10. Oriol Vinyals, Igor Babuschkin, Junyoung Chung, Michael Mathieu, Max Jaderberg, Wojtek Czarnecki, Andrew Dudzik, Aja Huang, Petko Georgiev, Richard Powell, Timo Ewalds, Dan Horgan, Manuel Kroiss, Ivo Danihelka, John Agapiou, Junhyuk Oh, Valentin Dalibard, David Choi, Laurent Sifre, Yury Sulsky, Sasha Vezhnevets, James Molloy, Trevor Cai, David Budden, Tom Paine, Caglar Gulcehre, Ziyu Wang, Tobias Pfaff, Toby Pohlen, Dani Yogatama, Julia Cohen, Katrina McKinney, Oliver Smith, Tom Schaul, Timothy Lillicrap, Chris Apps, Koray Kavukcuoglu, Demis Hassabis, and David Silver. Alphastar: Mastering the real-time strategy game starcraft ii. `https://deepmind.com/blog/alphastar-mastering-real-time-strategy-game-starcraft-ii/`, 2019. [Accessed: 2022.02.21].

11. Marco Dorigo, Dario Floreano, Luca Maria Gambardella, Francesco Mondada, Stefano Nolfi, Tarek Baaboura, Mauro Birattari, Michael Bonani, Manuele Brambilla, Arne Brutschy, Daniel Burnier, Alexandre Campo, Anders Lyhne Christensen, Antal Decugniere, Gianni Di Caro, Frederick Ducatelle, Eliseo Ferrante, Alexander Förster, Javier Martinez Gonzales, Jerome Guzzi, Valentin Longchamp, Stephane Magnenat, Nithin Mathews, Marco Montes De Oca, Rehan O'Grady, Carlo Pinciroli, Giovanni Pini, Philippe Rétornaz, James Roberts, Valerio Sperati, Timothy Stirling, Alessandro Stranieri, Thomas Stützle, Vito Trianni, Elio Tuci, Ali Emre Turgut, and Florian Vaussard. Swarmanoid: A novel concept for the study of heterogeneous robotic swarms. *IEEE Robotics and Automation Magazine*, 20(4):60–71, 2013.

12. Ankur Deka and Katia Sycara. Natural Emergence of Heterogeneous Strategies in Artificially Intelligent Competitive Teams. pages 13–25, 2021.

13. US Department of Defense. Directive 3000.09. `https://www.esd.whs.mil/portals/54/documents/dd/issuances/dodd/300009p.pdf`, 2012. [Accessed: 2022.02.21].

14. John Holland. *Hidden Order: How Adaptation Builds Complexity.* Addison-Wesley Publishing Company, Reading, MA, 1995.

15. John H Holland. *Signals and boundaries: Building blocks for complex adaptive systems.* MIT Press, 2012.

16. Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach.* Prentice Hall, Upper Saddle River, N.J., 3 edition, 2010.

17. Stan Franklin and Art Graesser. Is it an agent, or just a program?: A taxonomy for autonomous agents. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 1193(February 2014):21–35, 2015.

18. A. M. Turing. Computing machinery and intelligence. *Machine Intelligence: Perspectives on the Computational Model*, pages 1–28, 2012.

19. Marvin Minsky. *The Society of Mind.* Simon Schuster, New York, 1986.

20. Arthur L Samuel. Some Studies in Machine Learning Using the Game of Checkers. *IBM Journal of Research and Development*, 3(3):210–229, 1959.

21. Murray Campbell, A. Joseph Hoane, and Feng Hsiung Hsu. Deep Blue. *Artificial Intelligence*, 134(1-2):57–83, 2002.

22. Melanie Mitchell. *Artificial Intelligence: A Guide for Thinking Humans.* Picador, New York, 2019.

23. D Silver, J Schrittwieser, K Simonyan, I Antonoglou Nature, and Undefined 2017. Mastering the game of Go without human knowledge. *Nature*, 550(7676):354, 2016.

24. Rodney A Brooks. Elephants Don't Play Chess. *Robotics and Autonomous Systems*, 6:3–15, 1990.

25. R Menzel and U Müller. Learning and Memory in Honeybees : From Behavior to Neural Substrates. *Annual Review of Neuroscience*, 19:379–404, 1996.

26. Dario Floreano, Francesco Mondada, Andres Perez-uribe, and Daniel Roggen. Evolution of Embodied Intelligence. Technical report, Swiss Federal Institute of Techology, Lausanne, 2004.

27. Brian G. Woolley and Gilbert L. Peterson. Unified behavior framework for reactive robot control. *Journal of Intelligent and Robotic Systems: Theory and Applications*, 55(2-3):155–176, 2009.

28. Francois Chollet. On the Measure of Intelligence. Technical report, Google, Inc., 2019.

29. Yoav Shoham and Kevin Leyton-Brown. *Multiagent systems: Algorithmic, Game-Theoretic, and logical foundations*. 2008.

30. John H Holland. Studying complex adaptive systems. (November 2005):1–8, 2006.

31. Thomas Recchia, Jae Chung, and Kishore Pochiraju. Performance of heterogeneous robot teams with personality adjusted learning. *Biologically Inspired Cognitive Architectures*, 7:87–97, 2014.

32. Rinde R.S. van Lon and Tom Holvoet. When do agents outperform centralized algorithms?: A systematic empirical evaluation in logistics. *Autonomous Agents and Multi-Agent Systems*, 31(6):1578–1609, 2017.

33. Rinde Van Lon, Eliseo Ferrante, Ali Turgut, Tom Wenseleers, Vanden Berghe, Rinde Van Lon, Eliseo Ferrante, Ali Turgut, Tom Wenseleers, and Vanden Berghe. Measures of Dynamism and Urgency in Logistics. 2016.

34. L David Mech. Possible Use of Foresight, Understanding, and Planning by Wolves Hunting Muskoxen. *Arctic*, 60(2):145–149, 2007.

35. C. Muro, R. Escobedo, L. Spector, and R. P. Coppinger. Wolf-pack (Canis lupus) hunting strategies emerge from simple rules in computational simulations. *Behavioural Processes*, 2011.

36. Kao Shing Hwang, Harry Chia Hung Hsu, and Alan Liu. A homogeneous agent architecture for robot navigation. *Proceedings of 2003 International Conference on Neural Networks and Signal Processing, ICNNSP'03*, 1:310–315, 2003.

37. Truls Pedersen and Sjur Kristoffer Dyrkolbotn. Agents homogeneous: A procedurally anonymous semantics characterizing the homogeneous fragment of atl. In *International Conference on Principles and Practice of Multi-Agent Systems*, pages 245–259. Springer, 2013.

38. Eric Bonabeau, Marco Dorigo, and Guy Theraulaz. *Swarm Intelligence: From Natural to Artificial Systems*. Oxford University Press, New York, 1999.

39. Marco Dorigo, Vittorio Maniezzo, and A. Colorni. Positive feedback as a search strategy. Technical Report 91-016, Dipartimento di Elettronica, Politecnico di Milano, Milan, Italy, 1991.

40. Marco Dorigo and Gianni Di Caro. The Ant Colony Optimization meta-heuristic. In David Corne, Marco Dorigo, and Fred Glover, editors, *New Ideas in Optimization*, chapter 2, pages 11–32. McGraw-Hill, London, UK, 1999.

41. J. L. Deneubourg, S. Goss, N. Franks, and J. M. Pasteels. The blind leading the blind: Modeling chemically mediated army ant raid patterns. *Journal of Insect Behavior*, 2(5):719–725, 1989.

42. Philip Hingston and Graham Kendall. Enumerating knight's tours using an ant colony algorithm. In *2005 IEEE Congress on Evolutionary Computation, IEEE CEC 2005. Proceedings*, 2005.

43. Sabrina M. de Oliveira, Leonardo C.T. Bezerra, Thomas Stützle, Marco Dorigo, Elizabeth F. Wanner, and Sérgio R. de Souza. A Computational Study on Ant Colony Optimization for the Traveling Salesman Problem with Dynamic Demands. *Computers Operations Research*, page 105359, 2021.

44. Glenn A. Fink, Jereme N. Haack, A. David McKinnon, and Errin W. Fulp. Defense on the move: Ant-based cyber defense. *IEEE Security and Privacy*, 12(2):36–43, 2014.

45. Craig W. Reynolds. Flocks, herds, and schools: A distributed behavioral model. *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1987*, 21(4):25–34, 1987.

46. James Kennedy and Russell Eberhart. Particle swarm optimization. *ICNN'95-international conference on neural networks*, pages 1942–1948, 1995.

47. Christian Leonardo Camacho Villalón, Thomas Stützle, and Marco Dorigo. Grey Wolf, Firefly and Bat Algorithms: Three Widespread Algorithms that Do Not Contain Any Novelty. *Lecture Notes in Computer Science (including subseries*

Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 12421 LNCS(March):121–133, 2020.

48. S. Powell and E. Clark. Combat between large derived societies: A subterranean army ant established as a predator of mature leaf-cutting ant colonies. *Insectes Sociaux*, 51(4):342–351, 2004.

49. T. C. Schneirla and Gerard Piel. The Army Ant. 178(6):16–23, 1948.

50. L. David Mech. Alpha status, dominance, and division of labor in wolf packs. *Canadian Journal of Zoology*, 1999.

51. Ying Tan, editor. *Swarm Intelligence - Volume 1: Principles, current algorithms and methods*. The Institution of Engineering and Technology, London, 2018.

52. Marco A. Montes De Oca, Jorge Peña, Thomas Stützle, Carlo Pinciroli, and Marco Dorigo. Heterogeneous particle swarm optimizers. *2009 IEEE Congress on Evolutionary Computation, CEC 2009*, (January):698–705, 2009.

53. Sylwia Polberg, Marcin Paprzycki, and Maria Ganzha. Developing intelligent bots for the Diplomacy game. *2011 Federated Conference on Computer Science and Information Systems, FedCSIS 2011*, pages 589–596, 2011.

54. Tomer D. Ullman, Elizabeth Spelke, Peter Battaglia, and Joshua B. Tenenbaum. Mind Games: Game Engines as an Architecture for Intuitive Physics. *Trends in Cognitive Sciences*, 21(9):649–665, 2017.

55. David W. Aha, Matthew Molineaux, and Marc Ponsen. Learning to win: Case-based plan selection in a real-time strategy game. *Lecture Notes in Artificial Intelligence (Subseries of Lecture Notes in Computer Science)*, 3620:5–20, 2005.

56. Ben G. Weber, Michael Mateas, and Arnav Jhala. Building human-level AI for real-time strategy games. *AAAI Fall Symposium - Technical Report*, FS-11-01:329–336, 2011.

57. Federico Liberatore, Antonio M. Mora, Pedro A. Castillo, and Juan J. Merelo. Comparing Heterogeneous and Homogeneous Flocking Strategies for the Ghost Team in the Game of Ms. Pac-Man. *IEEE Transactions on Computational Intelligence and AI in Games*, 8(3):278–287, 2016.

58. David W. King, Jason M. Bindewald, and Gilbert L. Peterson. Informal team assignment in a pursuit-evasion game. *Proceedings of the 31st International Florida Artificial Intelligence Research Society Conference, FLAIRS 2018*, pages 32–37, 2018.

59. Ruck Thawonmas and Hiroshi Matsumoto. Automatic controller of Ms. Pac-Man and its performance: Winner of the IEEE CEC 2009 software agent Ms. Pac-Man competition. *Asia Simulation Conference 2009, JSST 2009*, (August), 2009.

60. David W. King and Gilbert L. Peterson. Decentralized control strategies for unmanned aircraft system pursuit and evasion. *IEEE Vehicular Technology Conference*, 2019-September, 2019.

61. Brood War API. `https://github.com/bwapi/bwapihttps://bwapi.github.io/`, 2011. [Accessed: 2022.02.21].

62. Oriol Vinyals, Timo Ewalds, Sergey Bartunov, Petko Georgiev, Alexander Sasha Vezhnevets, Michelle Yeo, Alireza Makhzani, Heinrich Küttler, John Agapiou, Julian Schrittwieser, John Quan, Stephen Gaffney, Stig Petersen, Karen Simonyan, Tom Schaul, Hado van Hasselt, David Silver, Timothy Lillicrap, Kevin Calderone, Paul Keet, Anthony Brunasso, David Lawrence, Anders Ekermo, Jacob Repp, and Rodney Tsing. StarCraft II: A New Challenge for Reinforcement Learning. 2017.

63. Anderson R. Tavares, Gianlucca Lodron Zuin, Héctor Azpúrua, and Luiz Chaimowicz. Combining genetic algorithm and swarm intelligence for task allocation in a real time strategy game. *Journal on Interactive Systems*, 8(1):1, 2017.

64. Navin K Adhikari. *Co-evolving Distributed Control for Heterogeneous Agents in RTS Games*. PhD thesis, 2019.

65. Mikayel Samvelyan, Tabish Rashid, Christian Schroeder De Witt, Gregory Farquhar, Nantas Nardelli, Tim G.J. Rudner, Chia Man Hung, Philip H.S. Torr, Jakob Foerster, and Shimon Whiteson. The StarCraft multi-agent challenge. *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS*, 4(NeurIPS):2186–2188, 2019.

66. Oriol Vinyals, Igor Babuschkin, Wojciech M. Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H. Choi, Richard Powell, Timo Ewalds, Petko Georgiev, Junhyuk Oh, Dan Horgan, Manuel Kroiss, Ivo Danihelka, Aja Huang, Laurent Sifre, Trevor Cai, John P. Agapiou, Max Jaderberg, Alexander S. Vezhnevets, Rémi Leblond, Tobias Pohlen, Valentin Dalibard, David Budden, Yury Sulsky, James Molloy, Tom L. Paine, Caglar Gulcehre, Ziyu Wang, Tobias Pfaff, Yuhuai Wu, Roman Ring, Dani Yogatama, Dario Wünsch, Katrina McKinney, Oliver Smith, Tom Schaul, Timothy Lillicrap, Koray Kavukcuoglu, Demis Hassabis, Chris Apps, and David Silver. Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature*, 575(7782):350–354, 2019.

67. Jack Shulimson, Leonard A. Blasiol, Charles R. Smith, and David A. Dawson. *US Marines in Vietnam: The Defining Year 1968*. Headquarters, US Marine Corps, Washington, DC, 1997.

68. Ulysses Simpson Grant et al. *Memoirs and Selected Letters: Personal Memoirs of US Grant, Selected Letters 1839-1865*, volume 50. Library of America, 1990.

69. Basil H. Liddell Hart. *Strategy*. Meridian, 2 edition, 1991.

70. John F. C. Fuller. *The Conduct of War, 1789-1961*. Minerva Press, 1961.

71. David W. King and Gilbert L. Peterson. The Emergence of Division of Labor in Multi-Agent Systems. *International Conference on Self-Adaptive and Self-Organizing Systems, SASO*, 2019-June:107–116, 2019.

72. William S. Lind. *Maneuver Warfare Handbook*. Routledge, 1985.

73. United States Marine Corps Headquarters. MCWP 3-11.2: Marine Rifle Squad. `https://www.marines.mil/Portals/1/Publications/MCWP%203-11.2%20Marine%20Rifle%20Squad.pdf`, 2002. [Accessed: 2022.02.21].

74. U.S. Army Corps of Engineers. Composite air wing from mountain home air force base, idaho, usa. in the wing are f-15 fighters, f-16 fighters, a kc-135 tanker, and a b1-b bomber. `https://commons.wikimedia.org/wiki/File:Composite_air_wing_USAF_Mountain_Home_AFB.jpg`, 1998. [Accessed: 2022.01.31].

75. United States Army Headquarters. Fm 3-96: Brigade combat team. `https://armypubs.army.mil/ProductMaps/PubForm/FM.aspx`, 2021. [Accessed: 2022.02.21].

76. Matthew Nelson, Fleming Larsen, and Pavel Savara. RoboCode. `https://github.com/robo-code/robocode`, 2022. [Accessed: 2022.01.31].

77. Jackie O'Kelly and J. Paul Gibson. RoboCode Problem-Based Learning: A non-prescriptive approach to teaching programming. *Proceedings of the Conference on Integrating Technology into Computer Science Education, ITiCSE*, (June):125–129, 2006.

78. Yehonatan Shichel, Eran Ziserman, and Moshe Sipper. GP-robocode: Using genetic programming to evolve robocode players. *Lecture Notes in Computer Science*, 3447:143–154, 2005.

79. Jacob Eisenstein. Evolving Robocode Tank Fighters. (October), 2003.

80. Jin Hyuk Hong and Sung Bae Cho. Evolution of emergent behaviors for shooting game characters in robocode, 2004.

81. António Rebelo, Fábio Catalão, João Alves, and Goreti Marreiros. Prototyping Teams of Affective Agents in Robocode. 15(1), 2015.

82. United States Army Headquarters. ADP 3-90: Offense and Defense. `https://armypubs.army.mil/ProductMaps/PubForm/ADP.aspx`, 2019. [Accessed: 2022.02.21].

83. United States Marine Corps Headquarters. MCRP 12-10B.1: Military Operations on Urbanized Terrain (MOUT). `https://www.marines.mil/News/Publications/MCPEL/Electronic-Library-Display/Article/900535/mcrp-12-10b1-formerly-mcwp-3-353/`, 2018. [Accessed: 2022.02.21].

84. George S. Patton. *War As I Knew It.* Houghton Mifflin, Boston, 1947.

85. Vago Muradian and James Mattis. Gen. James Mattis QA. `https://smallwarsjournal.com/index.php/blog/gen-james-mattis-qa`, 2010. [Accessed: 2022.01.31].

86. Sun Tzu and Lionel Giles. *The Art of War.* Project Gutenberg, Urbana, Illinois, 1994.

Acronyms

ACO Ant Colony Optimization. 10, 11

AI Artificial Intelligence. 2, 5, 6, 7, 21, 81

API Application Programming Interface. 16

APM Actions per Minute. 17

CAS Complex Adaptive System. 1, 8, 9

DoD Department of Defense. 3

GWO Grey Wolf Optimizer. 11

ITAA Informal Task Assignment Algorithm. 15

LCS Learning Classifier System. 80

MCDP Marine Corps Doctrinal Publication. 19

PSO Particle Swarm Optimization. 10, 11

RTS Real-Time Strategy. 15, 17, 19

SI Swarm Intelligence. 13

SMAC StarCraft Multi-Agent Challenge. 16, 17, 80

UBF Unified Behavior Framework. 22, 25, 32

USMC United States Marine Corps. iv, 19, 44, 60, 67

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704–0188

1. REPORT DATE *(DD–MM–YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From — To)*
25–03–2022	Master's Thesis	Jul 2020 — Mar 2022

4. TITLE AND SUBTITLE

Performance of Heterogeneous Multi-Agent Systems
with Applications in Combined Arms

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Wilson, Robert J., Captain, 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-MS-22-M-074

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Intentionally Left Blank

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

DISTRIBUTION STATEMENT A:
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Multi-agent systems show great potential for solving problems in complex and dynamic domains. Such systems comprise multiple individual entities called agents. Agents possessing the same behavior or physical form are called homogeneous while agents which differ in these respects are termed heterogeneous. The overall behavior of the system emerges from the many interactions of its component agents. Most multi-agent systems research to date focuses on systems of homogeneous agents, but recent work suggests that heterogeneous agents may improve system performance in certain tasks. This research examines the impact of heterogeneity on multi-agent system effectiveness and investigates the application of multi-agent systems to the complex problem of combined arms warfare. Hundreds of multi-agent teams are evolved and evaluated on their ability to complete certain objectives. Results show that combined arms tactics can emerge from the interactions of simple, decentralized agents, indicating that future research in this domain may prove valuable to the military art of combined arms warfare.

15. SUBJECT TERMS

Multi Agent Systems, Heterogeneous, Homogeneous, Distributed Control, Robotics, Evolution

16. SECURITY CLASSIFICATION OF: a. REPORT	b. ABSTRACT	c. THIS PAGE	17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON / 19b. TELEPHONE NUMBER *(include area code)*
U	U	U	UU	105	Lt Col David W. King, AFIT/ENG (937) 255-3636 x4370; david.king@afit.edu