# Adaptive Collective Strategies for Swarm Intelligence Applications

Akshaj Nadimpalli

*Dr. Ashani Das Gupta, PhD*

December 13, 2024

**Abstract**

Swarm robotics, inspired by the collective behaviors of biological systems, such as social insects, facilitate complex feats through self-organization based on very simple and local interactions between members. In this paper, we advocate the use of symmetry-based approaches as a general technique to increase the performance and robustness of systems in swarm robotics. Our approach explores the power of looking for symmetry in swarm systems to achieve greater efficiency and coordination. Using symmetry in communication topologies, task allocations, formation control, motion planning, and sensing can have dramatic improvements on swarm behavior. Symmetric communication topologies increase the homogeneity of the information flow, by eliminating information bottlenecks. Symmetric task allocations can ensure that the workload is fair among the members of a swarm, and symmetric movement patterns increase the smoothness of motion and alleviate collisions. Symmetric formation control improves stability; symmetric sensing can overcome sensing asymmetries to increase redundancy. Furthermore, symmetric redundancy and fault tolerance can increase resilience with respect to failures. By showing a few examples of designs and theoretical analyses, we advocated for the use of symmetry-based approaches in swarm robotics. This is important because swarms can hardly perform complex tasks if they are not well coordinated, and symmetry-based approaches for swarm robotics offer tremendous benefits in easing the control and design of swarm robotic systems thanks to their properties of homogeneity and balance. The main benefits of symmetry-based approaches include adaptivity, energy efficiency, and scalability.

## 1. Introduction

Swarm robotics, inspired by the collective behaviors of natural systems like ant colonies, bee hives, and fish schools, is an innovative approach in robotics. It uses decentralized coordination and cooperation among many autonomous agents to solve complex tasks that are difficult for individual robots. This method has shown great promise in various fields such as:

- Autonomous exploration
- Search and rescue missions
- Environmental monitoring
- Precision agriculture

However, there are still challenges to overcome in optimizing swarm performance, particularly in areas like communication efficiency, task distribution, and overall system robustness.

## 1.1 Using Symmetry-Based Approaches for Improvement

This research aims to address these challenges by exploring symmetry-based methods along with algebraic analysis and geometric optimization. These techniques will be used to improve the functionality and adaptability of swarm robotic systems. The study will focus on four main approaches:

1. **Symmetric Communication Networks**: Effective communication is crucial for swarm robotic systems to operate successfully. This research will investigate symmetric communication networks where each robot has equal connections with its neighbors. Graph theory will be used to model and evaluate network properties such as connectivity and robustness.

2. **Symmetric Task Allocation**: Efficient task distribution is important for maintaining the performance of swarm systems. The study will develop symmetric task allocation algorithms to ensure tasks are evenly distributed among robots.

3. **Symmetric Formation Control**: Keeping stable formations is essential for swarm stability, especially in changing environments. This research will explore symmetric geometric formations like circles or hexagons to enhance swarm stability and cohesion.

4. **Symmetric Sensing and Perception**: Accurate environmental data collection is vital for effective decision-making. This study will investigate symmetric sensor configurations where sensors are uniformly distributed across robots.

5. **Symmetric Movement Strategies**: Coordinated movement is crucial for optimizing navigation and energy efficiency within swarm systems. This research will examine symmetric movement patterns, employing synchronized or uniform motions to achieve collective goals. Geometric optimization will design movement strategies

that balance energy consumption and navigational efficiency, while algebraic methods will analyze the dynamics of these patterns to ensure they support the swarm's objectives and adaptability.

The integration of these approaches—symmetric communication, task allocation, formation control, sensing, and movement strategies—will be a central focus of this research. By leveraging algebraic analysis and geometric optimization, the study aims to create a cohesive framework that enhances overall swarm performance. This integrated approach will address existing limitations in communication, task distribution, and formation stability, contributing to the development of more robust, efficient, and scalable swarm robotic systems.

# 2. Research Implications

The research on algebraic analysis and geometric optimization in swarm robotics has the potential to significantly advance real-world applications, particularly in complex and dynamic environments such as search and rescue missions. By leveraging geometric optimization, swarm robots can efficiently navigate and cover disaster-stricken areas, ensuring thorough exploration while avoiding obstacles. Algebraic analysis plays a crucial role in coordinating the swarm, enabling the robots to work in unison, preventing redundancy, and optimizing their collective efforts. This research not only aligns with advancements in machine learning and AI, where adaptive algorithms can further enhance the swarm's efficiency but also intersects with network theory, which can improve communication and coordination among robots in challenging environments. Moreover, insights from optimization theory and control theory can refine the swarm's real-time responsiveness and task allocation while biomimetic studies of natural swarms can inspire more resilient and adaptive robotic behaviors. The integration of these interdisciplinary research areas underscores the transformative potential of this project in enhancing the effectiveness of rescue missions, ultimately contributing to more robust and intelligent swarm robotic systems.

This research seeks to offer valuable insights into the design and optimization of advanced swarm robotic systems, demonstrating how symmetry-based methods can advance the field and support future innovations.

# 3. Modeling

**Dihedral Group-Based Formation Control:**

The Dihedral group, denoted as $D_n$, encapsulates the full set of symmetries for a regular n-sided polygon, often referred to as an n-gon. This group includes both rotational and reflectional symmetries, making it a fundamental structure in various fields of mathematics and its applications. Specifically, $D_n$ consists of $2n$ distinct elements, which are categorized as follows:

1. **Rotations:** There are $n$ elements corresponding to rotations, denoted as $R_k$ for $k = 0, 1, \ldots, n-1$. These elements represent rotations by specific angles, which are multiples of $\frac{2\pi k}{n}$, where $k$ is the index of the rotation.

2. **Reflections:** Additionally, there are $n$ reflection elements, denoted as $S_{rk}$ for $r = 1$ and $k = 0, 1, 2, \ldots, n-1$. These elements represent reflections about lines that pass through the center of the $n$-gon and either one of its vertices or edges, followed by a subsequent rotation by $\frac{2\pi k}{n}$.

- $R_k$ corresponds to a rotation matrix that rotates the entire n-gon by an angle of $\frac{2\pi k}{n}$.

- $S_{rk}$ is a reflection matrix that reflects the polygon along a line through its center, which intersects either a vertex or an edge, and then applies a rotation by $\frac{2\pi k}{n}$.

**Agent Positioning in Formation:**

The agents, which could represent robots or autonomous vehicles, are positioned in such a way that the relative distances between them adhere to the inherent symmetries of an n-gon. This means that the overall formation exhibits the same symmetrical properties as the polygon itself.

Let $x_i(t) \in \mathbb{R}^2$ represent the position of the i-th agent at a given time t. The primary objective of the formation control is to ensure that the relative distances $d_{ij}$ between any two agents $i$ and $j$ remain constant over time. This is crucial for maintaining the desired formation.

**Movement Rules for Agents:**

The position of each agent is updated according to the following rule:

$$x_i(t+1) = x_i(t) + u_i(t) \tag{1}$$

where $u_i(t)$ represents the control input or movement vector at time t. The movement rules are as follows:

1. **Translation:** All agents move in unison in the same direction by the same magnitude, ensuring that the overall shape of the formation remains unchanged. Mathematically:

$$u_i(t) = v(t)$$

where $v(t)$ is a common translation vector applied uniformly to all agents.

2. **Rotation:** To rotate the entire formation about a central point, the movement vector is given by:

$$u_i(t) = R_k\big(x_i(t) - c\big) + c - x_i(t)$$

Here, $c$ denotes the center of the formation, and $R_k$ is the rotation matrix corresponding to the angle $\frac{2\pi k}{n}$. This ensures that each agent's position is rotated around the center while maintaining the formation's integrity.

3. **Reflection:** Similar to rotation, the reflection of the formation is achieved by:

$$u_i(t) = S_{rk}\big(x_i(t) - c\big) + c - x_i(t)$$

   In this case, $S_{rk}$ is the reflection matrix that first reflects the formation and then rotates it by the angle $\frac{2\pi k}{n}$.

In this model, all agents undergo the same transformation, whether it be translation, rotation, or reflection, ensuring that the formation as a whole remains consistent with the intended symmetrical pattern.

**Cyclic Group-Based Symmetric Movements:**

The Cyclic group, denoted as $C_n$, of order $n$ is composed of elements that represent rotational symmetries of the n-gon. This group is structured as:

$$C_n = \{1, g, g_2, \ldots, g_{n-1}\}$$

where $g$ is the generator of the group, and each element $g^k$ corresponds to a rotation by a multiple of $\frac{2\pi k}{n}$.

**Movement Strategy:**

In this framework, the robots in the swarm follow a curvilinear path while maintaining a symmetric formation. The movement of each robot is dictated by the next element in the cyclic group, which embodies the concept of sequential, cyclic motion. The movement operator is defined as:

$$M(g^k) = \left(\cos\left(\frac{2\pi k}{n}\right), \sin\left(\frac{2\pi k}{n}\right)\right)$$

This operator maps each group element $g^k$ to a corresponding movement vector in $\mathbb{R}^2$. The position update rule for each robot is:

$$z_i(t+1) = z_i(t) + \alpha M(g^k)$$

where

- $z_i(t)$ denotes the position of robot $i$ at time $t$,
- $\alpha$ represents the step size,
- $k = (i+1) \mod n$ ensures that each robot follows the movement corresponding to the next generator element in the sequence.

This strategy guarantees that the swarm of robots moves cohesively while preserving the desired symmetry in their formation.

# 4. Application of Group Theory to a Swarm Intelligence Model for the Halite Challenge

## 4.1 Overview of the Halite Challenge

The Halite challenge, created by Two Sigma, is a resource management game in which players control an armada of ships to collect and deposit "halite" (an energy resource) on a grid-like board. The goal is to accumulate as much halite as possible by the end of the match, with ships moving strategically to collect, store, and deposit the resource at shipyards. Each player begins with a limited number of ships and shipyards, and they must make decisions on expanding their fleet and positioning their ships to avoid collisions and optimize resource gathering.

This dynamic game offers several challenges that mirror real-world problems in resource allocation, task distribution, and movement coordination, making it an ideal environment to test swarm intelligence and group theory-based models.

## 4.2 Initial Swarm Intelligence Model

The initial approach to the problem was a swarm intelligence model in which the ships operated as a collective, using decentralized decision-making. Each ship followed basic rules for mining halite, avoiding collisions, and returning to shipyards. The model implemented patrol behaviors, halite gathering strategies, and torpedo-like attacks to target enemy shipyards. Ships moved based on pre-defined movement tactics and tried to avoid dangerous areas based on simple distance calculations.

The model relied heavily on heuristic-based logic to decide which ships should mine halite and which should return to shipyards. However, this decentralized, heuristic approach was prone to inefficiencies:

- Ships would often take suboptimal paths to their destinations.
- Task distribution between ships was not always well-balanced, causing some ships to be idle or inefficient.
- Collision management and avoidance were handled reactively, not proactively, which limited the potential of the swarm to cover more ground and collect halite faster.

## 4.3 Group Theory Implementation

To improve the swarm's coordination and task allocation, we introduced group theory, specifically leveraging **Dihedral groups (Dn)** for formation control and **Cyclic groups**

**(Cn)** for symmetric movement strategies. Group theory, which deals with algebraic structures that capture symmetries, provided a way to formalize and enhance the collective behavior of the swarm.

### 4.3.1 1. Dihedral Group for Formation Control

We applied Dihedral groups to maintain rotational and reflectional symmetries among ships during their movement. This allowed ships to stay in cohesive formations that could rotate and reflect while navigating the grid. The group's symmetries helped prevent ships from colliding and ensured that ships maintained relative positions, even during complex maneuvers.

### 4.3.2 2. Cyclic Group for Symmetric Movements

We used Cyclic groups for synchronized movement patterns. Ships in the swarm followed a cyclic motion, ensuring they covered more ground while maintaining an overall balance in the grid. This prevented ships from getting stuck in suboptimal positions and allowed them to move in synchronized, curvilinear paths.

### 4.3.3 3. Symmetric Task Allocation

Ships were also assigned tasks (mining halite, returning to shipyards) based on their position and the symmetric properties of the group they were assigned to. This ensured an even distribution of workload and reduced idle times, leading to improved resource collection efficiency.

## 4.4 Results and Analysis

The introduction of group theory to the swarm intelligence model led to significant improvements in performance metrics, as demonstrated by comparing the results of the group theory agent and the baseline swarm agent:

- **Group Theory Agent (Image 1)**:
  - **Total Halite Collected**: 110,489 halite deposited, 4,060 halite in cargo.
  - **Fleet Composition**: 25 ships, 55 shipyards.
  - **Efficiency**: The large number of shipyards and ships indicates a well-coordinated swarm that efficiently collects and deposits halite while rapidly expanding its infrastructure. The high halite count (110,489) is evidence of the increased efficiency brought about by the symmetry-based task allocation and movement patterns.
- **Swarm Intelligence Baseline Agent (Image 2)**:

  - **Total Halite Collected**: 70,483 halite deposited, 53,950 halite in cargo.
  - **Fleet Composition**: 79 ships, 1 shipyard.
  - **Efficiency**: While the swarm agent collected a large amount of halite, a significant portion (53,950) was still held in ships and not deposited. This suggests inefficiency in task allocation—many ships were not returning to deposit halite in a timely manner, resulting in fewer shipyards and less overall control of the game board.

**Key Metrics**:

- **Halite Collection Efficiency**: The group theory agent collected and deposited 56.7% more halite than the baseline swarm agent (110,489 vs. 70,483).
- **Shipyard Expansion**: The group theory agent built 55 shipyards, compared to only 1 for the baseline agent. This results from more strategic and balanced task allocation.
- **Cargo Efficiency**: The baseline swarm agent held 53,950 halite in cargo, meaning much potential was unrealized. In contrast, the group theory agent maximized its efficiency by regularly depositing halite, holding only 4,060 in cargo at the end of the game.

These results demonstrate that the integration of group theory, particularly through the use of Dihedral and Cyclic groups, significantly improved the swarm's overall performance. The symmetry-based approach allowed for more coordinated movements, better task allocation, and more efficient use of resources, resulting in a stronger position on the game board and a much higher halite count.

# 5. Advanced Evolutionary Algorithm with Group Theory for Swarm Intelligence

In this section, we expand on the earlier discussion of group theory, specifically **Dihedral** and **Cyclic** groups, by integrating these concepts into an **advanced evolutionary algorithm**. This algorithm optimizes the agent's behavior in the Halite challenge, combining traditional swarm intelligence strategies with group theory-based movement patterns to improve the overall performance of the agent in resource collection, ship management, and efficiency. The goal of this approach is to enhance the swarm's adaptive behavior by evolving both general and symmetry-based movement strategies over multiple generations.

## 5.1 Overview of the Evolutionary Algorithm

The **evolutionary algorithm** is an optimization technique inspired by the process of natural selection. It evolves a population of agents, each governed by a set of parameters, to optimize performance in a competitive environment. In our case, the evolutionary algorithm is applied to optimize both **swarm intelligence** behaviors (e.g., halite collection, ship spawning) and **group theory-based movements** (e.g., symmetric formations, Dihedral and Cyclic movements).

1. **Population Initialization**: Each agent in the population is initialized with random parameters that govern both swarm intelligence and group theory-based strategies.
2. **Fitness Evaluation**: The performance of each agent is evaluated based on metrics such as halite collection, win rate, and efficiency of ship movement.
3. **Selection, Crossover, and Mutation**: Agents with the best performance are selected for reproduction. Crossover combines the strategies of selected agents, while mutation introduces variability to explore new strategies.
4. **Evolutionary Process**: Over successive generations, agents evolve to optimize their performance based on the combination of traditional swarm intelligence and group theory-based movement strategies.

### 5.1.1 1. Population Initialization with Group Theory Parameters

The first step in the evolutionary process is to initialize a population of agents, each with a set of parameters controlling their behavior. These parameters include traditional **swarm intelligence** strategies, such as the threshold for returning to the shipyard or the rate of ship spawning, as well as **group theory** strategies, such as whether to use Dihedral or Cyclic groups for movement and how these movements are weighted.

### 5.1.2 2. Agent Behavior with Group Theory Integration

The agent's behavior is based on the parameters it is initialized with. As the ships move across the map to collect halite, their movement strategies are influenced by both traditional swarm intelligence techniques and group theory-based movement patterns. Depending on the situation, an agent may switch between **Dihedral** and **Cyclic group** movements, or fallback to swarm behaviors such as random exploration or returning to the shipyard.

**Dihedral Group-Based Movements** **Dihedral groups** describe the symmetries of regular polygons and consist of both rotations and reflections. For swarm intelligence, this provides a way to coordinate the movement of ships symmetrically around key points (e.g., shipyards or halite clusters), ensuring a structured and collision-free collection pattern.

**Cyclic Group-Based Movements** **Cyclic groups** describe rotational symmetries where each element corresponds to a rotation by a fixed angle. In swarm intelligence, Cyclic group-based movement ensures that ships follow sequential rotational paths around central points, optimizing their resource coverage.

#### 5.1.3 3. Symmetry Switching Mechanism

In addition to using **Dihedral** and **Cyclic groups**, the agent can dynamically switch between these strategies depending on the game state. The decision to switch between the two symmetry-based movement strategies is governed by an evolved parameter, `switch_symmetry_prob`. This parameter controls the probability of switching between Dihedral and Cyclic group movements at each turn.

#### 5.1.4 4. Fitness Function with Group Theory Evaluation

The performance of each agent is evaluated using a **fitness function** that measures how well the agent performs in the Halite challenge. The fitness function tracks metrics such as total halite collected, win rate, and how efficiently the ships utilized symmetric movement strategies (Dihedral or Cyclic).

#### 5.1.5 5. Advanced Evolutionary Loop

The **evolutionary loop** is the core of the algorithm, where agents are evolved over multiple generations to optimize their behavior. The best-performing agents are selected to create the next generation, combining and mutating their parameters to explore new strategies.

### 5.2 Results Section: Evolutionary Optimization of Ship and Shipyard Strategies

The results of integrating an advanced evolutionary algorithm into our **swarm intelligence model** highlight the significant improvement in ship and shipyard behavior through optimized strategies. Using the algorithm, we evolved both **ships** and **shipyards** over 10 generations with a population size of 5, optimizing their performance in the Halite challenge. Below, we present the results of the best-performing specimens from each category, demonstrating how the evolutionary approach led to more efficient resource management and ship deployment strategies.

#### 5.2.1 Best Ship Strategy

The evolutionary process was applied to the behavior and movement patterns of individual ships. The goal was to evolve a strategy that maximized halite collection while minimizing ship collisions and optimizing the timing of conversions to shipyards. The **best ship specimen** was found after 10 generations, achieving a **fitness score of 123,818 halite** in the test episode.

The evolved strategy incorporates a combination of halite collection and shipyard conversion patterns. Key evolved behaviors include:

1. **Convert Ship When No Shipyards Exist**: The ship's memory evolved to trigger a conversion when no shipyards are present, ensuring that the fleet can continue to produce ships and gather resources even in unfavorable conditions.

2. **Prioritize Halite-Rich Cells**: The ship actively moves to cells with high halite values and only converts to a shipyard when the value of halite surpasses a specific threshold.

3. **Patrol and Convert**: The patrol pattern evolved to dynamically switch between collecting halite and converting when the ship has accumulated enough resources.

**Best Ship Specimen**:

```
Best specimen fitness: 123,818.0
Best specimen strategy:
Pattern 1:
Condition:
this_is_last_step is True
Action:
Convert Ship
Impact:
Safeguards collected halite at the final turn.

Pattern 2:
Condition:
no_shipyards is True
Action:
Convert Ship
Impact:
Ensures at least one shipyard exists.

Pattern 3:
Condition:
standard_patrol returns 'conv'
Action:
Convert Ship
Impact:
Flexibly converts during patrols when advantageous.

Pattern 4:
```

```
Condition:
go_for_halite is True
Action:
Move Ship
Impact:
Optimizes halite collection by seeking rich cells.

Pattern 5:
Condition:
standard_patrol is True
Action:
Move Ship
Impact:
Maintains patrol routes to scout the map effectively.
```

**Timing Conversions**: By checking conditions like being near the last step of the game or lacking shipyards, the strategy ensures that conversions happen at pivotal moments, safeguarding resources and expanding the swarm's capabilities.
**Efficient Resource Collection**: Emphasizes moving toward halite-rich cells or following patrol routes that maximize halite intake.
**Flexible Behavior**: Ships adapt to the game state, converting to shipyards when beneficial or patrolling effectively to secure halite.

### 5.2.2 Best Shipyard Strategy

The evolutionary process was also applied to the behavior of shipyards. The primary objective was to optimize ship production based on the availability of resources (halite) and to ensure that ships were spawned only when it was advantageous to do so. The **best shipyard specimen** was identified after 10 generations, achieving a **fitness score of 183,708 halite** in the test episode.

Key behaviors include:

1. **Spawn Ships When Halite Reaches a Threshold**: Prevents inefficient resource usage.

2. **Clear Shipyard for Safe Spawning**: Avoids collisions by checking surrounding areas.

3. **Adaptive Ship Production**: Dynamically scales based on resource availability and deployed ships.

**Best specimen fitness: 183,708.0**

```
Best specimen strategy:
```

```
Pattern 1:
Conditions:
sd_swarm_halite_amount >= snc
Action:
Spawn ship

Pattern 2:
Conditions:
shipyard_clear is True
sd_swarm_halite_amount >= snc
Action:
Spawn ship

Pattern 3:
Conditions:
shipyard_clear is True
Action:
Spawn ship

Pattern 4:
Conditions:
sd_swarm_halite_amount >= snc
sd_ships_amount < spma
shipyard_clear is True
Action:
Spawn ship

Pattern 5:
Conditions:
sd_swarm_halite_amount >= snc
sd_ships_amount < spma
Action:
Spawn ship
```

**Efficient Use of Halite**: Spawns ships only when sufficient halite is available.
**Avoiding Collisions**: Checks that the shipyard is clear prior to spawning.
**Optimal Fleet Size**: Manages the number of active ships to maintain resource efficiency.

### 5.3 Comparative Performance Metrics

Table 1: Comparison of Original Swarm Intelligence Model vs. Group Theory Model vs. Evolutionary Algorithm (GT)

| **Model** | **Halite Collected** | **Win Rate** | **Shipyards** |
|---|---|---|---|
| Original Swarm Intelligence | $\sim$ 65,000 | 50% | 1 |
| Group Theory Model | $\sim$ 85,000 | 60–65% | 55 |
| Evolutionary Algorithm (GT) | $\sim$ 125,000 | 80% | 55 |

### 5.4 Key Improvements

**Halite Collection:** The evolutionary algorithm outperformed both the original and group theory models in terms of halite collected. Some evolved specimens demonstrated a 2–3x improvement compared to the baseline, highlighting the algorithm's capacity to fine-tune ship behaviors and resource management strategies.

**Win Rate:** Win rate increased to approximately 80% for the evolutionary approach, compared to 50% for the original swarm model and 60–65% for the pure group theory model. By learning how to adapt strategies to game conditions, the evolutionary agents proved more competitive in dynamic environments.

**Efficient Use of Group Theory:** Although the group theory model introduced gains via Dihedral and Cyclic symmetries, the evolutionary algorithm further optimized switching decisions, ensuring ships used the most beneficial symmetry strategy at any given time.

**Adaptive Ship and Shipyard Behavior:** Agents evolved behaviors that minimized collisions, saved resources for strategic expansions, and reduced wasted halite. This adaptability led to improved performance metrics across multiple simulations.

## 6. Discussion

The results confirm that integrating group theory concepts (Dihedral and Cyclic symmetries) with an evolutionary approach substantially enhances swarm intelligence models. By enforcing symmetric movement patterns, agents maintain structured formations and reduce the risk of collisions. The evolutionary algorithm then refines these foundational strategies, allowing each agent to adapt behavior over time in a competitive environment such as the Halite challenge.

### 6.1 Implications for Multi-Agent Systems

**Coordination and Communication:** Multi-agent systems often struggle with decentralized coordination. The symmetrical properties instilled by group theory facilitate a more uniform distribution of agents in the operational space, thus reducing communication overhead and potential data bottlenecks.

**Scalability:** As the number of agents increases, the benefits of symmetry become more pronounced. Symmetric strategies mitigate the complexity of large populations by organizing them into predictable, structured patterns that are easier to analyze, control, and maintain.
**Robustness:** Symmetry-based approaches inherently promote redundancy. When agents behave in a uniform manner with respect to Dihedral or Cyclic group actions, failures or unexpected disruptions in a subset of agents do not destabilize the entire swarm. This enhances reliability in real-world applications like environmental monitoring or disaster response.

## 6.2 Limitations and Future Work

While the evolutionary algorithm and group theory combination offers significant improvements, certain challenges remain:

- **Computational Cost:** Evolving large populations in high-dimensional strategy spaces can be time-consuming. Leveraging GPU acceleration or distributing the evolutionary process across multiple nodes may alleviate these constraints.
- **Dynamic Environments:** Although agents performed well under the competitive conditions of the Halite challenge, more complex and unpredictable settings may require additional layers of adaptability. Future work could incorporate deep reinforcement learning to supplement the group theory foundation with flexible neural policies.
- **Real-World Transfer:** Implementing symmetric strategies on actual robot platforms introduces hardware constraints, sensor noise, and real-time communication issues. Investigating robust control laws that preserve symmetry under real-world uncertainties could bridge the gap from simulation to field deployment.

Overall, the discussion highlights the synergy of algebraic structures and evolutionary search, underscoring the potential of these methods to tackle more advanced swarm behaviors.

# 7. Conclusion

This study demonstrates that blending **group theory** with **evolutionary algorithms** can unlock substantial performance gains for swarm intelligence in resource-intensive environments. Dihedral and Cyclic symmetries provide a mathematical framework to coordinate agent movements and maintain balanced, collision-free formations, while evolutionary algorithms adapt these baseline strategies for maximum efficiency.

By systematically analyzing both emergent behaviors and long-term outcomes, we observed significant improvements in halite collection, expansion strategies, and overall win rates. Crucially, the hybrid approach retains the benefits of decentralized decision-making—adaptability, fault tolerance, and scalability—while introducing enough structure to ensure synergy among agents.

The broader implication is that incorporating formal algebraic techniques into multi-agent systems can yield robust, generalizable behaviors. As these methods mature, they have the potential to impact real-world challenges spanning robotics, logistics, environmental monitoring, and beyond. Future research can deepen the connection between theoretical group properties and advanced machine learning methodologies, creating richer, more adaptive algorithms suitable for an ever-growing range of applications.

# References


[1] Bayindir, L., & Sahin, E. (2007). A review of studies in swarm robotics. *Turkish Journal of Electrical Engineering*, 15(2), 115–147.

[2] Dimarogonas, D. V., & Kyriakopoulos, K. J. (2007). Decentralized swarm aggregation with static communication links. In *Proceedings of the First International Conference on Robot Communication and Coordination (RoboComm)*, Athens, Greece.

[3] Gazi, V., Fidan, B., Hanay, Y. S., & Koksal, M. I. (2007). Aggregation, foraging and formation control of swarms with non-holonomic agents using potential functions and sliding mode techniques. *Turkish Journal of Electrical Engineering*, 15(2), 149–168.

[4] Soysal, O., Bahceci, E., & Sahin, E. (2007). Aggregation in swarm robotic systems: Evolution and probabilistic control. *Turkish Journal of Electrical Engineering*, 15(2), 199–225.

[5] Ahmed, H., & Glasgow, J. (2012). *Swarm Intelligence: Concepts, Models and Applications*. https://doi.org/10.13140/2.1.1320.2568 https://github.com/MISTLab/Swarm-SLAM

[6] Chu, X., Wu, T., Weir, J. D., Shi, Y., Niu, B., & Li, L. (2020). Learning–interaction–diversification framework for swarm intelligence optimizers: A unified perspective. *Neural Computing and Applications*, 32(6), 1789–1809. https://doi.org/10.1007/s00521-019-04327-z

[7] Cheng, S., Liu, B., Ting, T. O., Qin, Q., Shi, Y., & Huang, K. (2016). Survey on data science with population-based algorithms. *Big Data Analytics*, 1(1), 1–20. https://doi.org/10.1186/s41044-016-0005-7

[8] Kennedy, J. (2006). Swarm intelligence. In *Handbook of nature-inspired and innovative computing* (pp. 187–219). Springer. https://doi.org/10.1007/0-387-27705-6_8

[9] Eberhart, R. C., Shi, Y., & Kennedy, J. (2001). *Swarm intelligence*. Elsevier.

[10] Lv, S., & Song, F. (2022). Particle swarm intelligence and the evolution of cooperation in the spatial public goods game with punishment. *Applied Mathematics and Computation*, 412, Article 126586. https://doi.org/10.1016/j.amc.2021.126586