A VISIT-COUNT-BASED COMPLETE FORAGING STRATEGY FOR ROBOT SWARMS

A Thesis

by

ARTURO GONZÁLEZ

Submitted in Partial Fulfillment of the

Requirements for the Degree of

Master of Science

Major Subject: Computer Science

The University of Texas Rio Grande Valley

May 2026

A VISIT-COUNT-BASED COMPLETE FORAGING STRATEGY FOR ROBOT SWARMS

A Thesis
by
ARTURO GONZALEZ

COMMITTEE MEMBERS

Dr. Qi Lu
Chair of Committee

Dr. Yifeng Gao
Committee Member

Dr. Bin Fu
Committee Member

Dr. Li Zhang
Committee Member

May 2026

ABSTRACT

Gonzalez, Arturo Y. A Visit-Count-Based Complete Foraging Strategy for Robot Swarms. Master of Science (M.S.) May 2026, 30 pp., 3 tables, 11 figures, 42 references.

Swarm robotics systems often rely on a balance between exploration and exploitation to perform tasks like Central Place Foraging. While exploitation methods such as pheromone trails and site fidelity are well-studied, the efficiency of the underlying random search exploration remains a challenge, frequently leading to redundant coverage and wasted time as multiple agents repeatedly search the same fruitless areas. This thesis introduces a memory-enhanced exploration strategy designed to improve the efficiency of random search in a swarm of simple robots.

In our proposed algorithm, each robot, constrained with limited memory, periodically logs its recent locations while exploring. This spatial data is shared at a central server, which aggregates the information into a collective "visit map" representing the swarm's exploration history. This map is then used to intelligently guide future exploration; robots probabilistically avoid choosing destinations in areas with high visit counts, thereby discouraging redundant searches. The primary goal is to evaluate whether this collective memory system enables the swarm to find and collect all resources, particularly the final few clusters, more rapidly than a baseline swarm employing a purely random search. By comparing the performance of both strategies, this research aims to demonstrate that even with delayed, centralized information and minimal agent memory, a shared understanding of explored territory can significantly reduce task completion time and improve the overall efficiency of swarm foraging. Extensive simulation experiments demonstrate that the proposed strategy consistently outperforms the centrally placed baseline foraging algorithm (CPFA). Compared to CPFA, the proposed method reduces the total collection time by up to 33% and improves collection efficiency by more than 48% during the final stage of the mission.

ACKNOWLEDGMENTS

I am grateful to Dr. Qi Lu and the MARS Lab for supporting me and my research journey since my undergraduate years and for providing funding and great growth opportunities. I would also like to acknowledge the financial support provided by the NSF Expand AI program (No. 2434916), the NSF CREST Center for Multidisciplinary Research Excellence in Cyber-Physical Infrastructure Systems (MECIS) (No. 2112650), and the NSF MSI program (No. 2318682).

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Page

CHAPTER I

INTRODUCTION

Background and Problem Statement

Swarm robotics is an emerging field inspired by the collective behavior of social insects such as ants and bees. Through simple local interactions and decentralized decision-making, robotic swarms can perform complex tasks such as exploration, mapping, and resource collection without centralized control. One of the most studied frameworks in this field is the *Central Place Foraging Algorithm* (CPFA), which models the behavior of foraging ants that search for, collect, and return resources to a central server (Hecker & Moses, 2015). While the CPFA has demonstrated effectiveness in enabling decentralized foraging, it exhibits a significant inefficiency during the exploration phase. Robots performing random searches frequently revisit previously explored, resource-poor areas, resulting in redundant coverage and longer collection times. This inefficiency becomes particularly problematic in large or sparse environments where the last few remaining resources require broad and efficient exploration.

In the standard CPFA, individual robots rely on probabilistic rules to balance exploration and exploitation, such as pheromone signaling (Sumpter & Beekman, 2003) and site fidelity (Beverly et al., 2009). However, these mechanisms alone do not provide sufficient spatial awareness to prevent overlapping exploration. As a result, many agents waste valuable time exploring the same unproductive regions. This limitation motivated the development of an improved system that incorporates spatial memory and collective data sharing to enhance the efficiency of random search behaviors in robotic swarms.

Proposed Solution and Research Goals

This research proposes a Grid-Based Complete Foraging Algorithm (GCFA) that augments traditional CPFA with a simple, scalable form of spatial awareness. Each robot periodically records its location during random search and stores a limited history of its most recent positions. Upon returning to the server, robots transmit these recorded coordinates to a centralized data structure—referred to as the *visit map*—which represents a discretized grid of the arena.

The central controller aggregates this spatial data to identify regions of high visit frequency, effectively marking unproductive zones. During subsequent foraging rounds, robots probabilistically avoid these areas, thereby reducing redundant exploration and improving overall search efficiency. While this design does introduce a central coordination layer, overall per-robot communication is minimal and robots are only coordinated when they reach the central server.

The primary goals of this research are therefore:

1. To design and implement a Grid-Based Complete Foraging Algorithm (GCFA) that incorporates spatial data sharing among robots.
2. To evaluate the performance of this algorithm against the baseline CPFA across multiple simulated environments.
3. To explore the scalability of grid-based memory mapping and its effect on computational efficiency and foraging performance.

Through these objectives, this study aims to contribute to the advancement of efficient, autonomous swarm behaviors that can operate in dynamic and large-scale environments, with applications in environmental monitoring, search-and-rescue operations, and distributed resource collection.

CHAPTER II

BACKGROUND AND RELATED WORK

The Central Place Foraging Algorithm (CPFA)

The *Central Place Foraging Algorithm* (CPFA) is a biologically inspired framework that models the collective foraging behaviors of social insects, particularly ants and bees, which search for and retrieve resources to a centralized server. In swarm robotics, the CPFA provides a decentralized method for coordinating multiple agents to explore, collect, and return resources without centralized control or global communication (Lu et al., 2016).

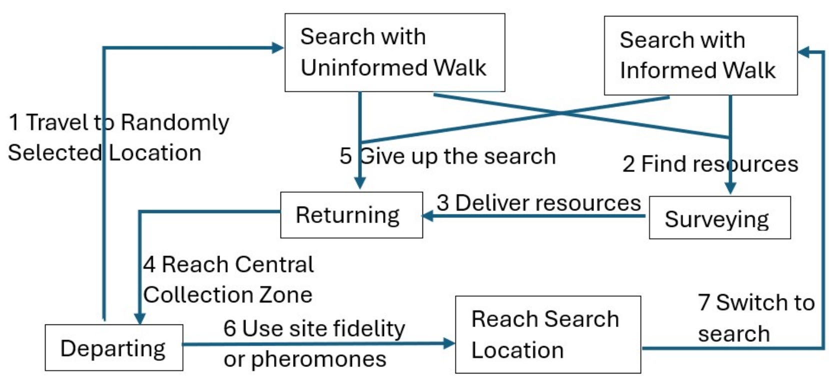

Figure 1: The flow chart of an individual robot's behavior and states in the CPFA

In the CPFA, each robot begins by performing a random search within a defined arena, as seen in figure 1. Upon finding a resource, the robot retrieves it and returns to the central server. During this return phase, the robot may deposit a virtual pheromone trail that encodes information about the resource's location (Campo et al., 2010). This trail decays over time but influences the movement of nearby robots, who probabilistically choose whether to follow pheromone cues (exploitation) or continue random searching (exploration).

The CPFA's success lies in its emergent coordination—robots indirectly communicate through the environment, allowing the swarm to collectively discover and harvest resources efficiently in dense clusters. However, while effective in early exploration, the algorithm often exhibits declining performance as resources become scarce. As the environment becomes depleted, robots spend increasingly more time exploring already-visited areas, leading to redundant searches and longer collection times. This inefficiency highlights the need for more intelligent exploration strategies that preserve CPFA's decentralized nature while improving spatial efficiency.

Exploration vs. Exploitation

A fundamental trade-off in foraging algorithms, including the CPFA, is the balance between *exploration* (searching new areas for undiscovered resources) and *exploitation* (revisiting areas known to contain resources). Efficient swarm performance requires maintaining this balance dynamically—favoring exploration when resources are unknown or sparse and shifting toward exploitation when resource clusters have been located.

The CPFA manages this balance probabilistically. Robots that find resources can engage in *site fidelity* (Beverly et al., 2009), returning to the same location with a certain probability, or follow pheromone trails deposited by other robots. Conversely, when no cues are available, robots perform *random walks* to explore new regions. However, random exploration lacks global awareness of which areas have already been visited by other agents, resulting in repeated searches of unproductive zones.

Several studies in swarm robotics have examined strategies to optimize this trade-off. Adaptive approaches modify exploration probability based on resource density or pheromone concentration, while others use local memory or environmental marking to bias movement toward unexplored areas. Despite these innovations, most methods remain reactive, responding to immediate sensory inputs without leveraging accumulated spatial knowledge. The work presented in this thesis builds on this literature by introducing a lightweight memory-sharing mechanism that enhances exploration efficiency without centralizing control.

Anti-Pheromones and Area Avoidance. The concept of *anti-pheromones*—virtual markers that discourage robots from visiting certain locations—has been explored as a method to reduce redundant exploration (Wagner et al., 1999). Unlike traditional pheromones that attract agents toward productive regions, anti-pheromones act as repellents, guiding robots away from zones that have already been extensively searched or deemed unproductive.

Research on anti-pheromone systems demonstrates that integrating avoidance cues can significantly improve spatial coverage and resource collection time (Burgard et al., 2005). For instance, algorithms that deposit negative pheromone markers in depleted zones can dynamically steer agents toward less-visited areas, improving the swarm's overall distribution. However, implementing such systems often introduces challenges related to parameter tuning, marker decay rates, and environmental noise.

The approach in this thesis aligns conceptually with the idea of anti-pheromones but replaces continuous environmental marking with a centralized, discrete *visit map*. Rather than diffusing negative cues throughout the environment, robots periodically record their positions and upload this data to the central server. The server aggregates these records into a shared occupancy grid that identifies highly visited regions. Robots then use this map indirectly—by probabilistically avoiding cells that exceed a visit threshold—thereby achieving area avoidance without relying on volatile pheromone dynamics.

This grid-based avoidance system provides a more stable and analyzable foundation for efficient exploration. It enables experimentation with granularity levels, computational trade-offs, and potential integration with machine learning models that predict resource-rich areas based on visitation patterns. Through this synthesis of stigmergic communication and collective spatial memory, the proposed algorithm extends traditional CPFA mechanisms toward more intelligent and informed swarm foraging.

CHAPTER III

METHODOLOGY

Simulation Environment

All experiments in this research were conducted using the ARGoS (Autonomous Robots Go Swarming) simulation platform (Pinciroli et al., 2012), which provides a flexible and efficient framework for large-scale multi-robot experiments. ARGoS allows for highly configurable arenas, robot models, and environmental conditions, enabling controlled and repeatable testing of swarm algorithms such as the Central Place Foraging Algorithm (CPFA).

The simulated environment consisted of a two-dimensional, enclosed arena representing the foraging field. Within this space, a swarm of 16 robots was deployed from a central collection zone and tasked with locating and retrieving dispersed resource clusters (Figure 2). To examine the scalability and efficiency of the proposed algorithm under different sizes, five primary arena configurations were evaluated: 8×8, 10×10, 12×12, 14×14, and 16×16 meter grids.

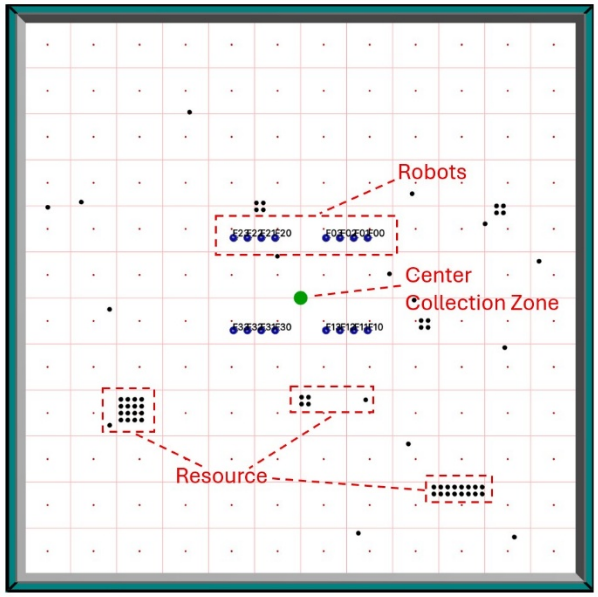

Figure 2: The grid of a 12 × 12 meter arena in ARGoS simulation

Baseline and Enhanced Foraging Algorithms

The baseline model implemented the standard Central Place Foraging Algorithm, which serves as a well-established framework for decentralized swarm foraging (Aggarwal et al., 2019). The CPFA consists of three primary behavioral phases: random search, resource collection, and return to server. During random search, robots go to a randomly selected target within the arena. Upon collection, the robot returns to the central server while optionally depositing a virtual pheromone trail that encodes the direction of the resource cluster.

Robots probabilistically decide between two strategies upon exiting the central server:

1. Exploitation – Following pheromone trails or returning to previously successful sites (site fidelity).

2. Exploration – Performing an uninformed random search to discover new resources.

While this system enables self-organization and efficient exploitation of dense resource clusters, it lacks global memory of previously explored regions. Consequently, multiple robots may redundantly explore the same unproductive zones, especially as resource density decreases over time.

To address the inefficiencies of redundant exploration, the Grid-Based Complete Foraging Algorithm (GCFA) introduces a lightweight, shared spatial memory mechanism integrated into the standard CPFA framework. This enhancement is designed to improve exploration efficiency without violating the decentralized principles of swarm intelligence.

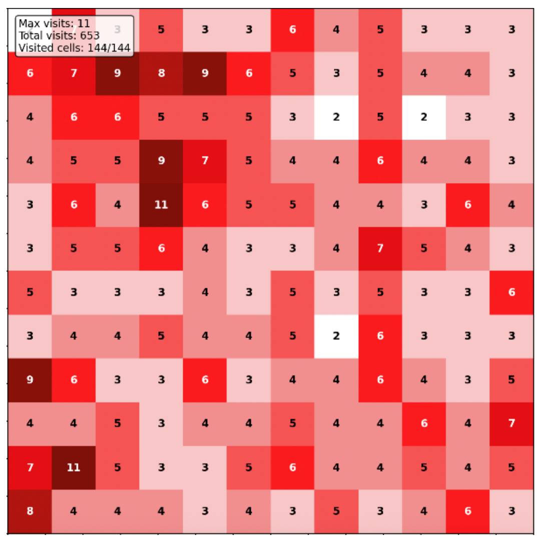

Figure 3: The counts of visits in each cell for the 12 × 12 meter arena

In the proposed approach, each robot maintains a local FIFO (First-In-First-Out) queue that stores its most recent visited locations during the random search phase. This efficient memory design follows ideas established by Lima and Oliveira (2017), who demonstrated that a short-term position memory list (inspired by Tabu Search in their paper) is sufficient to reduce redundant exploration in swarm foraging without requiring complex global memory. Locations are logged every 25 seconds. This interval balances two competing concerns: logging too frequently exhausts onboard memory and inflates communication overhead on each server return, while logging too infrequently produces a stale visit map that lags behind actual robot movement. The maximum memory capacity of 20 recent positions similarly reflects this trade-off: the queue must be large enough to describe meaningful spatial coverage during a search leg, yet it has to be small enough and scalable to remain feasible on resource-constrained hardware of the kind common in swarm platforms (Rubenstein et al., 2012). When a robot returns to the server, it uploads its stored location data to a centralized server, which maintains an $n \times n$ discretized grid representation of the arena. The server updates the grid-level visitation counts based on these reported locations (Figure 3). During subsequent search phases, the algorithm uses this aggregated map to actively bias exploration toward under-visited regions. The following behaviors were implemented in the enhanced algorithm:

- Periodic Spatial Logging: Robots record their positions every 25 seconds during uninformed search into a local queue of up to 20 locations.

- Server Communication: Upon return to the central server, robots transmit stored coordinates to the server, updating the central visit map.

- 3×3 Target Selection: The central server identifies individual minimum across the visit map. If there is more than one, the server breaks the tie by selecting the 3×3 block of cells with the minimum cumulative visit count and assigns this region to the departing robot (seen in the numbered cells of Figure 4).

- Systematic Local Search: The robot picks a random location in each of the 9 cells, visits them in a fixed counter-clockwise order, and performs a spiral search upon reaching each assigned cell (Fricke et al., 2016), demonstrated in the red, green, and blue paths of Figure 4.

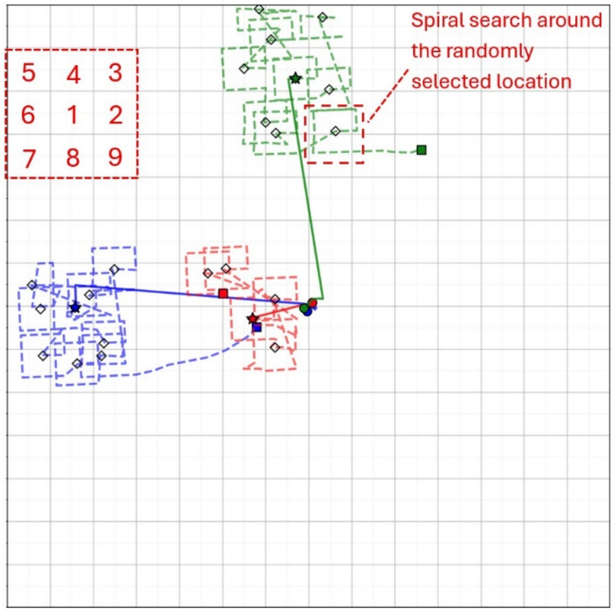

Figure 4: The illustration of the search pattern in selected regions

Experimental Design. To evaluate the effectiveness of the GCFA, a series of controlled simulations was conducted comparing it against the baseline CPFA under identical conditions. Each experiment was replicated across 30 trials. The primary performance metric was the total foraging time required for the swarm to collect all available resources. To thoroughly test the algorithm's robustness, the experiments explored the effects of varying:

- Number of Resources: Testing volumes of 16, 32, 48, 64, and 80 total resources.
- Resource Distributions: Evaluating performance across Clustered, Powerlaw, and Random distributions.
- Arena Size: Scaling the environment from 8×8 meters up to 16×16 meters.

Resource Distribution Types. There was three resource distribution types used in this research that reflect a range of real-world spatial patterns and represent different challenges for swarm foraging. Each distribution was generated using the ARGoS resource placement framework consistent with prior CPFA benchmark studies (Aggarwal et al., 2019; Hecker & Moses, 2015).

Random Distribution. In the Random distribution, resources are placed at uniformly sampled locations across the entire arena with no spatial structure. Each resource occupies an independently drawn position, resulting in a scattered, low-density spread with no clusters or gradients. This distribution represents the most challenging condition for pheromone-based exploitation strategies, because there is no resource density gradient to reinforce with trails. At the same time, it most directly rewards the GCFA's global coverage strategy. This is because resources are spread evenly, so a visit-count-based map that directs robots toward unvisited areas is well aligned with the task structure. This distribution serves as the primary benchmark and is used for all arena-size scalability experiments.

Power Law (Partially Clustered) Distribution. In the Power Law distribution, resources are placed in groups whose sizes follow a power law (heavy-tailed) statistical relationship. Concretely, a small number of large clusters contain many resources, while a larger number of smaller clusters contain few. Cluster centers are placed randomly across the arena, and individual resources are dispersed around each center. This pattern is ecologically motivated: natural food sources such as seed patches and prey aggregations commonly exhibit this kind of scale-free clustering (Crist & MacMahon, 1991). The Power Law distribution is the standard test condition used in CPFA benchmark studies (Hecker & Moses, 2015) because it combines both dense patches and sparse regions, rewarding algorithms that balance exploitation of known sites with broad exploration for undiscovered ones. The GCFA's visit-count map is expected to improve performance here by steering robots away from already-exhausted cluster locations and toward unexplored regions of the arena.

Clustered Distribution. In the Clustered distribution, all resources are concentrated into a small number of dense, tightly packed groups, each placed at a fixed location in the arena. Unlike the Power Law distribution, cluster sizes are all equal. This distribution is the most exploitable for pheromone-based strategies since once a robot discovers a cluster, it deposits a strong pheromone trail that reliably guides other robots to the same high-density location. The GCFA is expected to be at a disadvantage here relative to the baseline CPFA, because the visit-count map accumulates high values near the dense cluster sites, eventually steering robots away from the remaining resources still concentrated there. This tension between the GCFA's bias toward spatial diversity and the Clustered distribution's reward for repeated local exploitation represents the primary design trade-off explored in this thesis.

Quantitative results were tracked across the entire collection cycle, with specific analytical focus applied to the final 20% of the collection phase to measure the algorithm's impact on the "endgame slowdown". Table 1 summarizes the full set of experimental configurations used across all three experiment types.

Table 1: Experimental configurations used for testing different aspects, such as scalability with resources, endgame slowdown, and scalability with arena size.

Experiments	**I**	**II**	**III**
Arena Size	14 × 14	14 × 14	8 × 8, 10 × 10, 12 × 12, 14 × 14, 16 × 16
# of Resources	16, 32, 48, 64, 80	48	48
# of Robots	16	16	16

# of Runs	30	30	30
Resource Distributions	Clustered, Powerlaw, Random	Clustered, Powerlaw, Random	Random

CHAPTER IV

RESULTS

Flexibility with Different Numbers of Resources

To assess the flexibility of the algorithms, foraging performance was evaluated as the total number of resources increased from 16 up to 80. In exploration-heavy tasks, specifically within Random and Power law distributions, the performance gap between the algorithms widened significantly as the resource count increased.

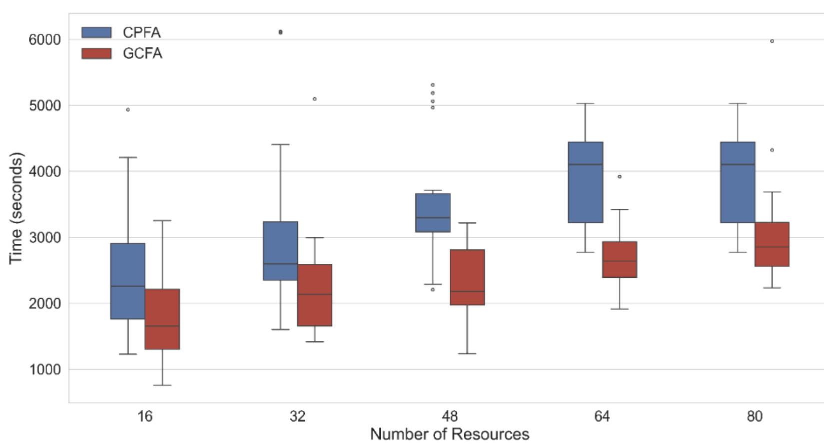

Figure 5: The time to complete the collection in the random distribution

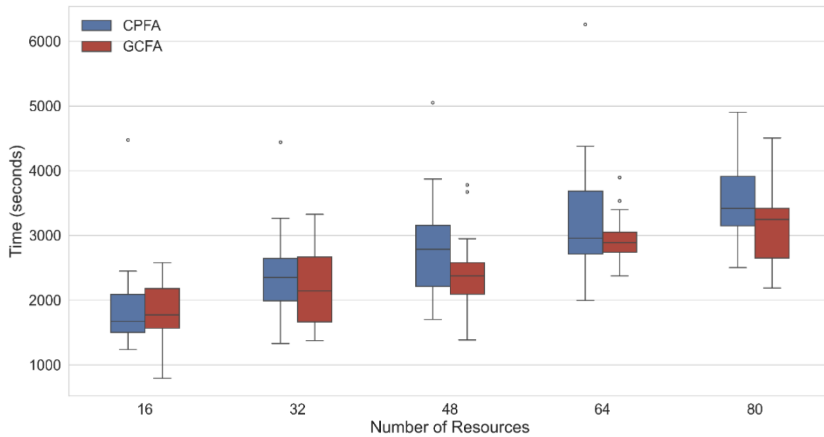

Figure 6: The time to complete the collection in Power law distribution

In the Random distribution, the performance gap at 16 resources was approximately 400 seconds, but at 80 resources, this gap increased to more than 800 seconds in favor of the GCFA (Figure 5). A similar trend was observed in the Power law distribution, where the performance gap grew from approximately 90 seconds at 16 resources to over 360 seconds at 80 resources (Figure 6). This demonstrates that the CPFA takes progressively more time in heavier foraging tasks, whereas the GCFA maintains a more linear relationship between resource count and collection time. At a baseline of 48 resources, the GCFA achieved an overall completion time improvement of 33.46% in the Random distribution and 14.11% in the Power law distribution compared to the CPFA. However, there was no significant difference in overall performance between the two algorithms in the highly exploitative Clustered distribution (Figure 7).

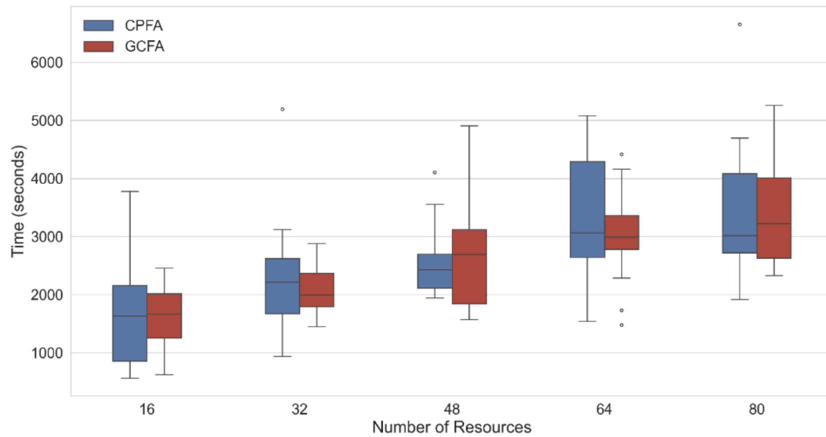

Figure 7: The time to complete collection in Clustered distribution

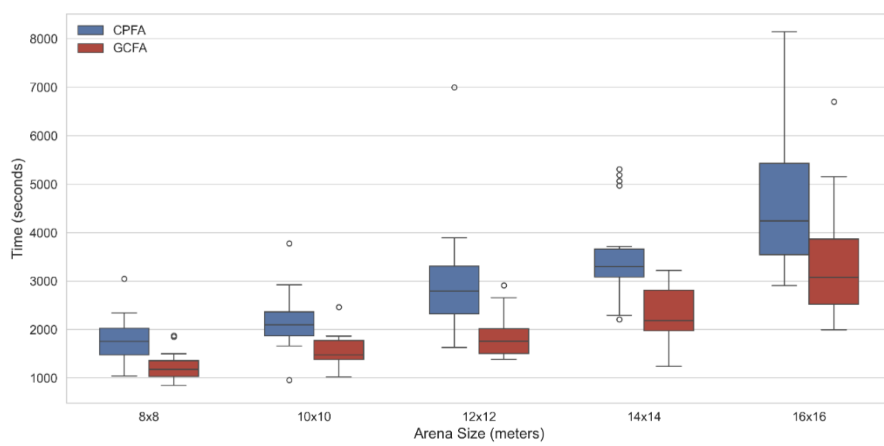

Figure 8: The time to complete the collection in different arena sizes

Robustness Across Different Arena Sizes

To determine how well the swarm scales in varied environments, the algorithms were tested in arenas ranging from 8x8 meters to 16x16 meters, using a Random distribution of 48 resources. The results indicate that the performance gap between the two algorithms increases (from 581 s to 1267 s across the five tested arena sizes) significantly as the arena size increases (Figure 8).

In the smallest 8x8 meter arena, the density of the robots relative to the search area allowed agents to cover the map quickly, resulting in a moderate time savings of 581 seconds for the GCFA (1222 seconds for GCFA vs. 1803 seconds for CPFA). However, as the arena quadrupled to 16x16 meters, the performance of the CPFA severely degraded, jumping to a completion time of 4564 seconds. In contrast, the GCFA completed the task in 3298 seconds, more than doubling the time saved to 1267 seconds (Figure 8; see Table 2 for a full breakdown by arena size). In large, sparse environments, the cost of redundantly searching previously cleared areas is high; the GCFA successfully minimizes this travel cost, proving to be a highly scalable solution.

Table 2: Time saved in different arena sizes (Random Dist., Resources = 48)

Arena Size (m)	GCFA	CPFA	Time Saved
8 × 8	1222	1803	**581**
10 × 10	1541	2165	**624**
12 × 12	1827	2973	**1146**
14 × 14	2355	3540	**1184**
16 × 16	3298	4564	**1267**

Analysis of the Final Collection Phase. To quantify the "endgame slowdown" commonly experienced in memoryless systems as resource density approaches zero, the collection timeline was broken down to analyze Interval A (the 80% to 90% collection phase) and Interval B (the final 90% to 100% collection phase).

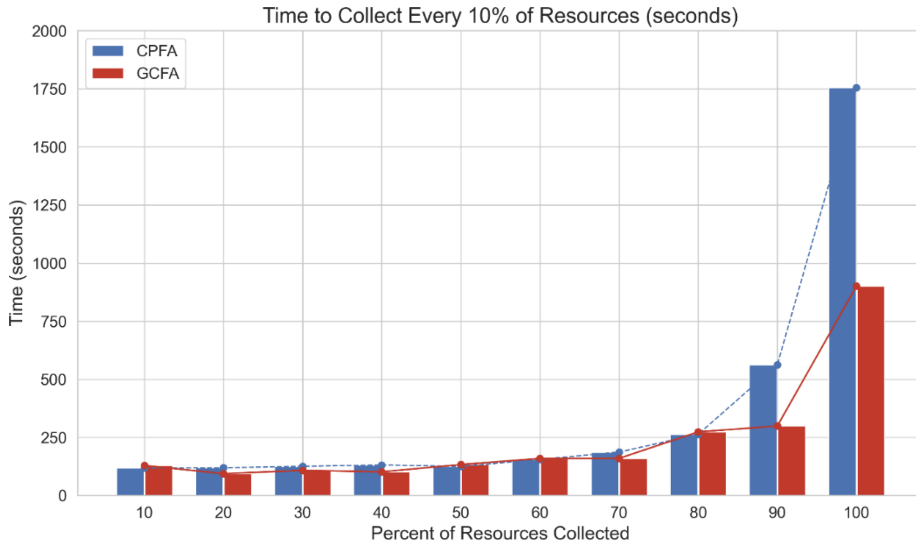

Figure 9: The time to collect every 10% of resources in Random distribution

In the Random distribution, the CPFA suffered a severe performance penalty during the final phase, with the duration spiking from 562.5 seconds in Interval A to 1755.2 seconds in Interval B (Figure 9). By retaining a memory of visited cells, the GCFA swarm effectively reduced its search space and completed the final Interval B in 900.8 seconds—an improvement of 48.68% over the CPFA. The Powerlaw distribution mirrored this trend; the GCFA completed Interval B in 774.8 seconds compared to the CPFA's 1199.2 seconds, yielding a 35.39% reduction in endgame time.

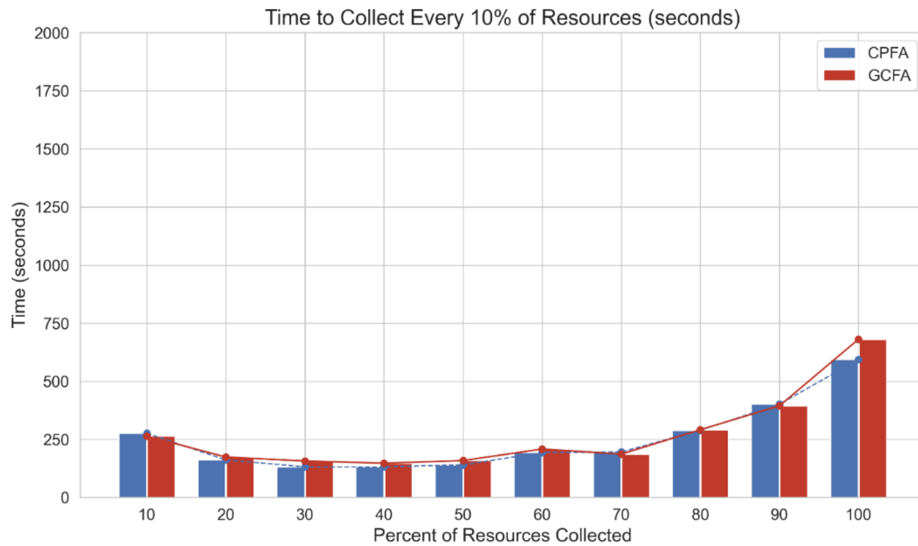

Figure 10: The time to collect every 10% of resources in Cluster distribution

Conversely, the Clustered distribution results highlighted a trade-off in the GCFA's design (Figure 11). The CPFA outperformed the GCFA in the final interval for clustered resources, completing it in 593.5 seconds compared to the GCFA's 680.7 seconds (a 14.69% decrease in efficiency). Because the final resources were located very close to previous finds within dense clusters, the CPFA's tendency to stay and search in a localized spot was highly advantageous. In this specific scenario, the GCFA's strategy to check unvisited areas backfired, forcing robots to travel to empty corners of the arena rather than persisting at known cluster sites. Table 3 summarizes the endgame interval performance across all three distributions.

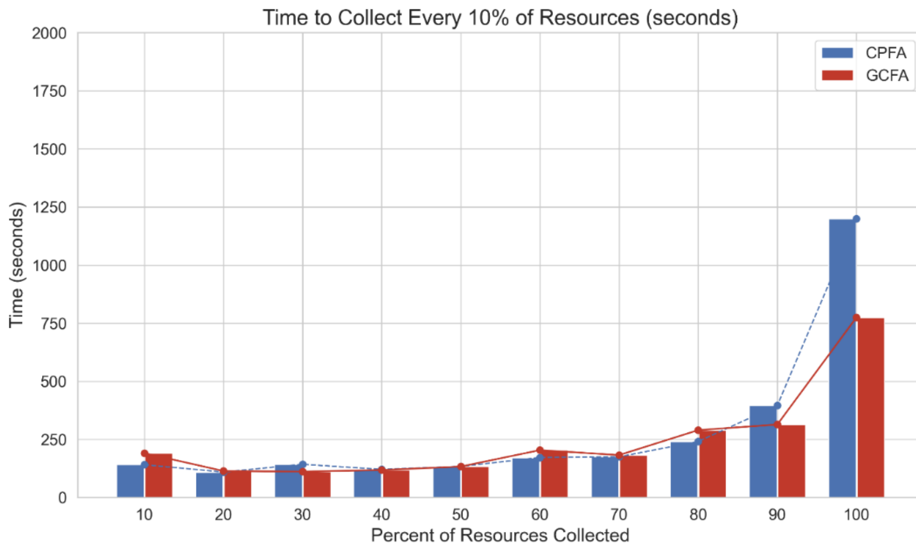

Figure 11: The time to collect every 10% of resources in Power law distribution

Table 3: Endgame Improvement CPFA vs GCFA (Random Dist., Resources = 48)

Config.	CPFA 80% → 90%	GCFA 80% → 90%	CPFA 90% → 100%	GCFA 90% → 100%	Endgame Improvement
Random	562.5	298.6	1755.2	900.8	**+48.68%**
Powerlaw	395.5	313.7	1199.2	774.8	**+35.39%**
Cluster	401.4	394.0	593.5	680.7	**-14.69%**

CHAPTER V

DISCUSSION AND CONCLUSION

Interpretation of Findings

The experimental results demonstrate that the Grid-Based Complete Foraging Algorithm (GCFA) significantly outperforms the canonical Central Place Foraging Algorithm (CPFA) in scenarios with high exploration requirements. By utilizing a shared grid-based map maintained by a central server, robots can proactively select unvisited or less-visited regions rather than relying on pure random search. This mechanism effectively alleviates the exponential time growth typically observed during the collection of the final 20% of resources, increasing efficiency by over 48% during this final phase. Furthermore, robustness evaluations confirm that the benefits of this grid-based approach increase alongside the arena size.

However, the results also highlight a critical performance trade-off in highly clustered environments, where the CPFA matched or outperformed the GCFA. While grid-based map exploration is superior for scattered resources, balancing broad exploration with local exploitation remains critical for clustered distributions. In the GCFA, cells near resource clusters quickly accumulate high visit counts, reducing their likelihood of being selected during subsequent uninformed searches. Consequently, any remaining resources hidden within these clusters become difficult to discover if they were not found earlier via pheromone waypoints.

Additionally, while the algorithm maintains scalability by limiting memory and communication throughput, it introduces specific computational constraints at the central server.

The running time required to select a target region for an uninformed search is $O(n^2)$, where n represents the number of grid cells. Specifically, sorting these cells by their visit counts requires $O(n^2)$ time, which dominates the overall computation. The number of cells directly dictates a trade-off: a larger n results in smaller cell sizes and a greater number of unvisited cells, which improves foraging efficiency. However, increasing n simultaneously leads to a higher computational cost.

Limitations and Future Work

The findings from this study lay the groundwork for several avenues of future research to refine stochastic search strategies for robots with limited capabilities. Future work will aim to extend this research by:

- Conducting a theoretical analysis of the relationship between the number of grid cells and the resulting computation cost.
- Investigating strategies to optimize the cell selection process to mitigate the sorting computation burden.
- Exploring communication between robots to construct local maps
- Validating the approach in physical swarm robotic platforms to assess real-world feasibility.
- Address the single point of failure from the nature of the central server.

Conclusion. This thesis proposed the Grid-Based Complete Foraging Algorithm (GCFA) to address the challenge of complete resource collection in autonomous robot swarms. By discretizing the unknown search region into a grid map maintained by a lightweight central server, the GCFA explicitly reduces redundant visits and accelerates late-stage collection. Operating under strict memory and computational constraints, robots share summarized trajectory data to generate a global estimate of exploration density, which intelligently biases future search patterns.

Extensive simulations demonstrate that this strategy reduces total collection time by up to 33% and improves endgame efficiency by more than 48% compared to the baseline CPFA. These results confirm that the GCFA is an effective and scalable solution for complete resource collection, proving that a centralized, grid-based spatial memory can serve as a highly effective general enhancement for stochastic swarm foraging.

REFERENCES

Khaldi, B., Harrou, F., Cherif, F., & Sun, Y. (2018). Self-organization in aggregating robot swarms: A DW-KNN topological approach. Biosystems, 165, 106–121.

Lima, D. A., & Oliveira, G. M. B. (2017). A cellular automata ant memory model of foraging in a swarm of robots. Applied Mathematical Modelling, 47, 551–572. https://doi.org/10.1016/j.apm.2017.03.021

Rubenstein, M., Ahler, C., & Nagpal, R. (2012). Kilobot: A low cost scalable robot system for collective behaviors. IEEE International Conference on Robotics and Automation (ICRA), 3293–3298.

Gauci, M., Chen, J., Li, W., Dodd, T. J., & Groß, R. (2014). Self-organized aggregation without computation. The International Journal of Robotics Research, 33(8), 1145–1161.

Nayak, S., Yeotikar, S., Carrillo, E., Rudnick-Cohen, E., Jaffar, M. K. M., Patel, R., Azarm, S., Herrmann, J. W., Xu, H., & Otte, M. (2020). Experimental comparison of decentralized task allocation algorithms under imperfect communication. IEEE Robotics and Automation Letters, 5(2), 572–579.

Arvin, F., Turgut, A. E., Bazyari, F., Arikan, K. B., Bellotto, N., & Yue, S. (2014). Cue-based aggregation with a mobile robot swarm: a novel fuzzy-based method. Adaptive Behavior, 22(3), 189–206.

Arvin, F., Turgut, A. E., Krajník, T., & Yue, S. (2016). Investigation of cue-based aggregation in static and dynamic environments with a mobile robot swarm. Adaptive Behavior, 24(2), 102–118.

Vardy, A. (2012). Accelerated patch sorting by a robotic swarm. 9th Conference on Computer and Robot Vision (CRV), 314–321.

Vardy, A., Vorobyev, G., & Banzhaf, W. (2014). Cache consensus: Rapid object sorting by a robotic swarm. Swarm Intelligence, 8, 61–87.

Qi, L., Hecker, J. P., & Moses, M. E. (2018). Multiple-place swarm foraging with dynamic depots. Autonomous Robots, 42, 909–926.

Llenas, A. F., Talamali, M. S., Xu, X., Marshall, J. A., & Reina, A. (2018). Quality-sensitive foraging by a robot swarm through virtual pheromone trails. Lecture Notes in Computer Science, 11172, 135–149.

Jin, B., Liang, Y., Han, Z., & Ohkura, K. (2020). Generating collective foraging behavior for robotic swarm using deep reinforcement learning. Artificial Life and Robotics, 25, 588–595.

Lee, D., Lu, Q., & Au, T.-C. (2021). Multiple-place swarm foraging with dynamic robot chains. IEEE International Conference on Robotics and Automation (ICRA), 11337–11342.

Kaminka, G. A., & Douchan, Y. (2025). Heterogeneous foraging swarms can be better. Frontiers in Robotics and AI, 11, 1426282.

Hecker, J. P., & Moses, M. E. (2015). Beyond pheromones: evolving error-tolerant, flexible, and scalable ant-inspired robot swarms. Swarm Intelligence, 9, 43–70.

Lu, Q., Hecker, J. P., & Moses, M. E. (2016). The MPFA: A multiple-place foraging algorithm for biologically-inspired robot swarms. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 3815–3821.

Fricke, G. M., Joshua, P. H., Antonio, D. G., Linh, T. T., & Melanie, E. M. (2016). A Distributed Deterministic Spiral Search Algorithm for Swarms. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 4430–4436.

Hecker, J. P., Carmichael, J. C., & Moses, M. E. (2015). Exploiting clusters for complete resource collection in biologically-inspired robot swarms. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 434–440.

Talamali, M. S., Bose, T., Haire, M., Xu, X., Marshall, J. A. R., & Reina, A. (2020). Sophisticated collective foraging with minimalist agents: a swarm robotics test. Swarm Intelligence, 14, 1935–3820.

Yamauchi, B. (1997). A frontier-based approach for autonomous exploration. IEEE International Symposium on Computational Intelligence in Robotics and Automation, 146–151.

Burgard, W., Moors, M., Stachniss, C., & Schneider, F. E. (2005). Coordinated multi-robot exploration. IEEE Transactions on Robotics, 21(3), 376–386.

Wagner, I. A., Lindenbaum, M., & Bruckstein, A. M. (1999). Distributed covering by ant-robots using evaporating traces. IEEE Transactions on Robotics and Automation, 15(5), 918–933.

Frazzoli, E., & Bullo, F. (2004). Decentralized algorithms for vehicle routing in a stochastic time-varying environment. 43rd IEEE Conference on Decision and Control, 3357–3363.

Pavone, M., Frazzoli, E., & Bullo, F. (2011). Adaptive and distributed algorithms for vehicle routing in a stochastic and dynamic environment. IEEE Transactions on Automatic Control, 56(6), 1259–1274.

Beverly, B. D., McLendon, H., Nacu, S., Holmes, S., & Gordon, D. M. (2009). How site fidelity leads to individual differences in the foraging activity of harvester ants. Behavioral Ecology, 20(3), 633.

Sumpter, D. J., & Beekman, M. (2003). From nonlinearity to optimality: pheromone trail foraging by ants. Animal Behaviour, 66(2), 273–280.

Jackson, D. E., Martin, S. J., Ratnieks, F. L. W., & Holcombe, M. (2007). Spatial and temporal variation in pheromone composition of ant foraging trails. Behavioral Ecology, 18(2), 444–450.

Campo, A., Gutiérrez, Á., Nouyan, S., Pinciroli, C., Longchamp, V., Garnier, S., & Dorigo, M. (2010). Artificial pheromone for path selection by a foraging swarm of robots. Biological Cybernetics, 103(5), 339–352.

Crist, T. O., & MacMahon, J. A. (1991). Individual foraging components of harvester ants: movement patterns and seed patch fidelity. Insectes Sociaux, 38(4), 379–396.

Pinciroli, C., Trianni, V., O'Grady, R., Pini, G., Brutschy, A., Brambilla, M., Mathews, N., Ferrante, E., Di Caro, G., & Ducatelle, F. (2012). ARGoS: a modular, parallel, multi-engine simulator for multi-robot systems. Swarm Intelligence, 6(4), 271–295.

Lu, Q. (2019). An Efficient Multiple-place Foraging Algorithm for Scalable Robot Swarms. Ph.D. thesis, The University of New Mexico.

Alshawi, M. A., & Shalan, M. B. (2017). Minimal Time Dynamic Task Allocation for a Swarm of Robots. International Journal of Mechanical Engineering and Robotics Research, 6(6), 481–487.

Aggarwal, A., Gupta, D., Vining, W. F., Fricke, G. M., & Moses, M. E. (2019). Ignorance is Not Bliss: An Analysis of Central-Place Foraging Algorithms. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 6510–6517.

Bharaswadkar, A., Vakilian, V., Thoms, B., & Isaacs, J. T. (2021). Congestion Strategies for Clustered Central Place Foraging. IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC), 185–190.

Marcolino, L. S., & Chaimowicz, L. (2009). Traffic control for a swarm of robots: Avoiding target congestion. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 1955–1961.

Marcolino, L. S., Passos, Y., De Souza, Á., Rodrigues, A., & Chaimowicz, L. (2016). Avoiding target congestion on the navigation of robotic swarms. Autonomous Robots, 41, 1297–1320.

dos Passos, Y. T., Duquesne, X., & Marcolino, L. S. (2022). Congestion control algorithms for robotic swarms with a common target based on the throughput of the target area. Robotics and Autonomous Systems, 159, 104284.

Vinicius, G. S., & Chaimowicz, L. (2011). Hierarchical congestion control for robotic swarms. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 4372–4377.

Soma, K., Vardharajan, V. S., Hamann, H., & Beltrame, G. (2023). Congestion and Scalability in Robot Swarms: A Study on Collective Decision Making. International Symposium on Multi-Robot and Multi-Agent Systems (MRS), 199–206.

Benhamou, S. (2004). How to reliably estimate the tortuosity of an animal's path: straightness, sinuosity, or fractal dimension?. Journal of Theoretical Biology, 229(2), 209–220.

Benhamou, S. (2006). Movement in a confined space: Estimating path tortuosity. Behavioral Ecology, 17(3), 619–629.

Garnier, L., Combe, M., Jost, C., & Theraulaz, G. (2013). Do ants need to estimate the geometrical properties of trail bifurcations to find an efficient route? A study using synthetic agents. PLOS Computational Biology, 9(3).

VITA

Arturo Gonzalez earned his Master of Science in Computer Science in May 2026 and his Bachelor of Science in Computer Science at the University of Texas at Rio Grande Valley in May of 2024. During undergraduate studies, Arturo discovered the MARS (Multiple Autonomous Robotic Systems) Lab. He joined in August 2023 and has since been contributing and learning under the guidance of Dr. Qi Lu. His experiences and lessons learned as a research assistant in this lab helped him take on other endeavors, such as internships in national labs like Argonne National Laboratory and industry settings like J.P. Morgan Chase. His personal email for any questions is arturoglz2004@gmail.com.