Scalable and Flexible Swarm Robotics for Constrained Foraging in ARGoS

Alexandre Le Mercier[1]

Université Libre de Bruxelles (ULB), Belgium
alexandre.le.mercier@ulb.be

Abstract. This paper addresses the constrained foraging problem in swarm robotics, where robots must collectively retrieve food items in an environment with physical obstacles between the source and the nest. Inspired by natural ant colonies, we investigate the role of stigmergy and self-organization in improving swarm performance. Two paradigms are compared: INDI-BOT, a baseline controller in which robots operate independently, and SWARM-BOT, a cooperative strategy that extends INDI-BOT with stigmergic signaling and flocking interactions.
We implemented both paradigms in the ARGOS multi-robot simulator and conducted 530 experiments to evaluate their scalability and flexibility. Results show that SWARM-BOT significantly outperforms INDI-BOT, particularly as swarm size increases, demonstrating the benefits of decentralized coordination. Furthermore, SWARM-BOT maintains basic functionality across different obstacle configurations, although its efficiency drops substantially in more cluttered environments.
We identify several limitations of the current approach, including sensitivity to floor reflectance and lack of adaptive exploration. We discuss how these could be addressed through more robust nest marking, coordinated transport strategies, and learning-based behaviors. Our findings highlight the effectiveness of simple bio-inspired mechanisms in distributed robotic systems and suggest promising directions for enhancing their robustness and generality.

Keywords: Swarm robotics · Constrained foraging · Stigmergy · Self-organization · ARGoS simulation

1 Introduction

Many animal species adopt swarm-based behaviors to solve tasks requiring complex cooperation. Ants, for instance, frequently face the *foraging* problem: locating food and returning it to the nest. In *constrained foraging*, physical obstacles separate the source from the nest. Ant colonies address this using *stigmergy* (leaving indirect environmental cues) and *self-organization* (coordination through local interactions), allowing them to succeed despite limited individual capabilities [1].

These mechanisms have inspired computational approaches such as *ant colony optimization* (ACO) [2]. In robotics, swarm systems are decentralized and scalable: performance improves with more agents, without central control or major reconfiguration. Their distributed nature also ensures fault tolerance, while complex behaviors like cooperative transport or path formation emerge from simple rules [3].

This study addresses constrained foraging using the SWARM-BOT paradigm, implemented in the ARGoS simulator [4]. To support reproducibility, the full setup is available on GitHub at `TortueSagace/swarm_bot`. Section 2 reviews related work, followed by Section 3 which introduces INDI-BOT, a non-cooperative baseline. Section 4 presents the cooperative SWARM-BOT strategy, and Section 5 analyzes results from 530 experiments. Section 6 discusses limitations and directions for future work. Finally, Section 7 concludes the paper.

2 Related Work

Constrained foraging has been widely studied in swarm robotics, where decentralized systems retrieve items in environments with obstacles. Early approaches relied on random walks and phototaxis, using simple stimuli such as light or proximity [5]. While inherently scalable, these strategies often suffer from redundant exploration and lack of coordination. To improve efficiency, bio-inspired methods such as virtual pheromone trails [6] and beacon-based path guidance [7] have been proposed. These stigmergic approaches enable swarms to converge on efficient paths, even in dynamic terrains.

Other strategies use direct communication (for example, range-and-bearing signals), improving reactivity and resource discovery in small groups [5]. Adaptive role allocation, such as response threshold models [8], can also mitigate interference in cluttered environments. Task partitioning, where subsets of robots specialize in phases of the transport chain, has proven effective in constrained domains [9].

3 Individual Footbots

This section describes the INDI-BOT paradigm, in which each ARGoS footbot[1] performs the foraging task independently, ignoring the presence of neighbors. INDI-BOT serves as a baseline for the SWARM-BOT paradigm presented in the next section, and is expected to perform poorly, thereby highlighting the value of cooperative behavior.

3.1 Sensors and Actuators

Each footbot uses an omnidirectional camera, proximity sensors, and ground sensors to update binary stimuli such as:

- `food_detected`, `neighbor_chasing`, `neighbor_carrying`, `nest_alert` based on blob colors (red, orange, green, purple);

[1] The footbot is the specific robot model used in ARGoS. Please refer to the ARGoS documentation for details.

- food_proximity, wall_proximity based on forward proximity readings;
- nest_detected based on a learned threshold of floor reflectance, updated with the darkest value seen so far.

Footbots are controlled via differential drive, a front gripper, and on-board RGB LEDs used for debugging. Motion is defined by FORWARD_SPEED and TURN_SPEED, with steering handled by a linear–angular mix. Timers enforce conditions like approach_timeout and carry_timeout.

3.2 Finite State Machine

INDI-BOT footbots follow the finite state machine (FSM) in Fig. 1. They start in WANDER_BOT, exploring until food is detected. Then, in APPROACH_BOT, they head toward the target. Upon reaching it, they enter CARRY_BOT, executing the following:

1. Move forward to slightly push the item;
2. If SHOULD_LOCK_POSITIVE is false (as in INDI-BOT), turn 180°, grab the item from behind, and drag it (DRAGGING_MARGIN);
3. Drop the item when nest_detected is triggered or dragging is complete;
4. Resume exploring with ignoring_food activated to avoid immediate re-engagement.

The robot's behavior after dropping depends on whether it is an F_CARRY_BOT or B_CARRY_BOT, which is selected stochastically via an annealing parameter $\alpha(t)$:

$$\alpha(t) = \alpha_{\text{target}} - (\alpha_{\text{target}} - \alpha_{\text{start}})\, e^{-t/\tau},$$

where $\tau =$ ALPHA_TAU steps. Low α favors exploration; high α biases return-like movement.

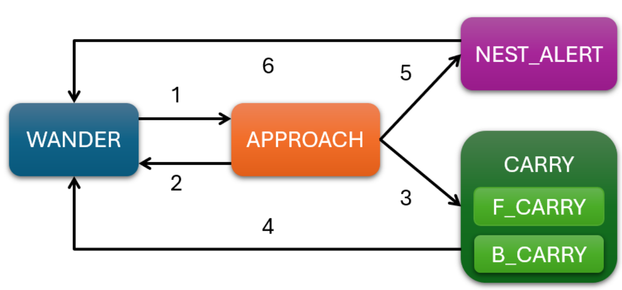

Fig. 1. Finite State Machine (FSM) of the robot controller. States include WANDER_BOT, APPROACH_BOT, and sub-states F_CARRY_BOT and B_CARRY_BOT within the CARRY_BOT phase. Transitions **(5)** and **(6)** are exclusive to the SWARM-BOT paradigm. Stimuli and conditions for each transition are listed in Table 1 for INDI-BOT and Table 2 for SWARM-BOT.

Table 1. Finite-state machine transitions in INDI-BOT.

ID	From State	To State	Triggering Stimulus or Condition
1	WANDER_BOT	APPROACH_BOT	food_detected and not ignoring_food
2	APPROACH_BOT	WANDER_BOT	approach_timeout
3	APPROACH_BOT	CARRY_BOT	food_proximity and food_detected
4	CARRY_BOT	WANDER_BOT	nest_detected or carry_timeout

4 Emergence of Swarm-based Cooperation

SWARM-BOT introduces cooperative features on top of INDI-BOT.

4.1 Stigmergy

To improve coordination, footbots in SWARM-BOT emit color-coded LED signals depending on their state. This provides short-range implicit communication:

(i) When food is detected, a robot turns orange, triggering `neighbor_chasing` in nearby `WANDER_BOT` bots. This prompts others to head toward the discovery.

(ii) During `CARRY_BOT`, robots turn green, activating `neighbor_carrying` and `ignoring_food` in others, preventing collisions over carried food.

(iii) Upon detecting the nest, a bot in the `APPROACH_BOT` state enters `NEST_ALERT_BOT` and emits a purple signal, triggering `nest_alert` and `ignore_food` in neighbors. This discourages robots from clustering near the nest and promotes dispersal.

4.2 Flocking

Flocking behavior mitigates disorganized exploration and collisions. A Lennard-Jones potential regulates inter-robot spacing, attracting distant neighbors and repelling close ones. Each bot computes a vector from nearby robots via the range-and-bearing sensor and probabilistically adjusts its heading (`FLOCKING_PROBA`). Flocking is active during `WANDER_BOT` and `APPROACH_BOT` (when following a neighbor), but disabled when heading towards food and during `CARRY_BOT` to avoid interference during transport.

5 Experimental Results and Discussion

A total of 530 foraging experiments were conducted in ARGoS, each lasting 6,000 time steps, to compare SWARM-BOT against INDI-BOT and evaluate the scalability and flexibility of the proposed swarm behaviors. For practical reasons, the footbots' speed was increased to 30 cm/s (instead of the default 10 cm/s) to ensure that the emergence of self-organizing behaviors could be observed within the fixed simulation time budget.

Table 2. Finite-state machine transitions in SWARM-BOT.

ID	From State	To State	Triggering Stimulus or Condition
1	`WANDER_BOT`	`APPROACH_BOT`	(`food_detected` or `neighbor_chasing`) and not `ignoring_food`
2	`APPROACH_BOT`	`WANDER_BOT`	`neighbor_carrying` or `approach_timeout` or `nest_alert`
3	`APPROACH_BOT`	`CARRY_BOT`	`food_proximity` and `food_detected`
4	`CARRY_BOT`	`WANDER_BOT`	`nest_detected` or `carry_timeout`
5	`APPROACH_BOT`	`NEST_ALERT_BOT`	`nest_detected`
6	`NEST_ALERT_BOT`	`WANDER_BOT`	`nest_alert_timeout`

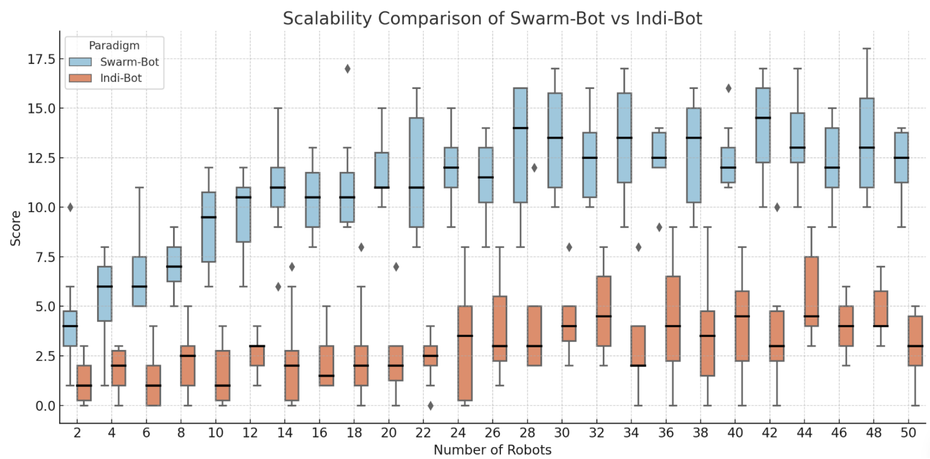

Fig. 2. Scalability comparison between SWARM-BOT and INDI-BOT across different swarm sizes (2 to 50 robots). Each boxplot shows the distribution of foraging scores over 10 trials. Each scalability test was conducted in the search space identified as "Scenario 1" in Fig. 3.

Figure 2 shows the results of the scalability tests performed in Scenario 1 (`foraging_s1`), comparing the individualist INDI-BOT paradigm with the co-operative SWARM-BOT controller. Across all swarm sizes, SWARM-BOT consistently outperforms INDI-BOT by a significant margin. The global average score of SWARM-BOT reaches 11.06, compared to only 3.14 for INDI-BOT. This demonstrates the clear advantage of stigmergic communication and decentralized self-organization mechanisms in distributed robotic systems. Additionally, the performance of SWARM-BOT increases with the swarm size up to approximately 20–30 robots, after which it saturates. This confirms that the strategy is scalable, although spatial congestion and coordination limits naturally appear in dense swarms.

Figure 3 evaluates the flexibility of the SWARM-BOT controller across four scenarios with different obstacle layouts, while keeping the control parameters unchanged. The average score in Scenario 1 is 9.00, compared to a significantly

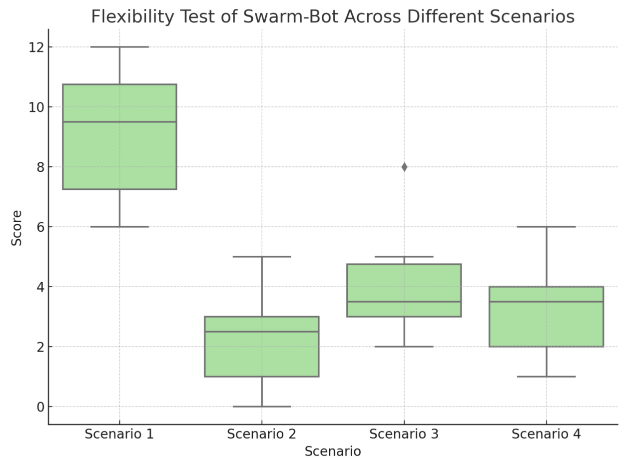

Fig. 3. Flexibility evaluation of Swarm-Bot across four different scenario layouts using the same control strategy. Each boxplot corresponds to the foraging performance of 10 robots in a distinct environment (`foraging_s1` through `foraging_s4`).

lower mean of 3.20 across Scenarios 2 to 4. While some performance drop is expected due to environmental variation, the discrepancy is particularly pronounced in Scenario 2, where robots struggle to maintain efficient exploration. This can be attributed to the increased total wall length in that configuration: as footbots in the `WANDER_BOT` state tend to follow arena edges, a more obstructed layout causes longer exploratory paths and higher dispersal time. These results underline the need for adaptive exploration mechanisms when generalizing to unseen environments.

6 Limitations and Further Work

While the proposed Swarm-Bot controller demonstrates significant scalability and partial flexibility, several limitations remain in its current form. First, nest detection is entirely based on ground reflectance levels, which can be sensitive to slight variations in floor color, shadows, or light diffusion. As a result, the inferred nest threshold may occasionally misclassify dark regions or lead to inconsistent drop locations across robots. Furthermore, despite the inclusion of a drop margin, some footbots still grab food items positioned at the edge of the nest, causing unintended removal of previously deposited items. A more robust approach would be to combine ground sensing with explicit feedback or local memory of successful drops. This echoes previous work on beacon-based navigation [7], which could be integrated to improve nest localization.

Another limitation arises in the `WANDER_BOT` state. Although footbots are equipped with flocking behavior during exploration, the interaction is stochastic and mostly suppressed through a probabilistic switch. This design choice prevents excessive aggregation and promotes dispersion. However, in highly obstructed layouts (e.g., Scenario 2), this can still lead to inefficient path selection, especially when robots remain near the periphery due to collision-based bouncing. Adaptive exploration policies encouraging centerward movement or adjusting flocking based on density could help. This aligns with adaptive response threshold models [8], which modulate behavior based on spatial and social feedback.

In the `CARRY_BOT` state, robots operate independently and do not coordinate their trajectories. This can lead to congestion, especially near the nest, where multiple robots may converge without awareness. While collisions are partially mitigated through reactive turns and early stops, the lack of coordination still limits throughput. Additionally, the grab sequence is not fully reliable: footbots may occasionally miss the food and return empty. Low-bandwidth coordination (e.g., obstacle signaling) could improve transport efficiency, building on prior work in task partitioning [9].

Finally, the system does not exploit long-term memory or learning. All behaviors are reactive, based solely on local stimuli and timers. Introducing memory, map-building, or reinforcement learning could enable smarter foraging, adaptive path refinement, and more resilient nest detection. Similar directions have been explored through virtual pheromone gradients [6], suggesting promising extensions.

These limitations suggest clear directions for future research, including robust nest marking, adaptive flocking, collaborative transport, and the integration of learning mechanisms. Many of these ideas are grounded in prior work on constrained foraging, as reviewed in Section 2.

7 Conclusion

This project addressed the constrained foraging problem by comparing two control paradigms: Indi-Bot, a non-cooperative baseline, and Swarm-Bot, a decentralized strategy using stigmergy and self-organization. Both were tested in the ARGoS simulator through 530 experiments assessing scalability and flexibility.

Results show that Swarm-Bot significantly outperforms Indi-Bot, especially as swarm size increases, confirming the benefits of implicit communication and distributed coordination. While Swarm-Bot retains moderate flexibility in varied environments, its performance drops in cluttered scenarios, highlighting the challenges of generalization.

Future work should explore more robust nest detection, better coordination during transport, and learning-based exploration. Such enhancements could improve adaptability in complex or dynamic environments.

References

1. Goss, S., Aron, S., Deneubourg, J.-L., Pasteels, J.M.: Self-organized shortcuts in the Argentine ant. *Naturwissenschaften* **76**(12), 579–581 (1989). https://doi.org/10.1007/BF00462870
2. Dorigo, M., Birattari, M., Stützle, T.: Ant Colony Optimization. *IEEE Computational Intelligence Magazine* **1**(4), 28–39 (2006). https://doi.org/10.1109/MCI.2006.329691
3. Dorigo, M., Birattari, M.: *INFO-H-414: Swarm Intelligence*. Université Libre de Bruxelles (2025). https://cs.ulb.ac.be/public/teaching/infoh414
4. Pinciroli, C. et al.: ARGoS: a modular, parallel, multi-engine simulator for multi-robot systems. *Swarm Intelligence* **6**(4), 271–295 (2012). https://doi.org/10.1007/s11721-012-0072-5
5. Pradhan, A., Amador, S., Winfield, A.F.T.: A Comparative Analysis of Collective Foraging Strategies. In: 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 3850–3857 (2020). https://doi.org/10.1109/IROS45743.2020.9341574
6. Talamali, M.S., Reina, A., Bose, T., Marshall, J.A.R., Garnier, S.: Bridging the gap between individual and collective decision-making in animal groups: an agent-based modeling approach. *Swarm Intelligence* **14**, 273–301 (2020). https://doi.org/10.1007/s11721-020-00178-z
7. Adams, B., Hecker, J.P., Christensen, H.I.: Efficient robot foraging using local communication and beacon robots. *Swarm and Evolutionary Computation* **80**, 101345 (2023). https://doi.org/10.1016/j.swevo.2023.101345
8. Castello, E., Yamamoto, T., Tuci, E., Groß, R.: Adaptive foraging under spatial constraints in robot swarms. *Swarm Intelligence* **10**(2), 117–147 (2016). https://doi.org/10.1007/s11721-016-0119-7
9. Ferrante, E., Duéñez-Guzmán, E.A., Turgut, A.E., Dorigo, M.: Self-organized task partitioning in swarms of robots. *Autonomous Agents and Multi-Agent Systems* **29**, 157–189 (2015). https://doi.org/10.1007/s10458-014-9253-4