

This version of the contribution has been accepted for publication, after peer review (when applicable) but is not the Version of Record and does not reflect post-acceptance improvements, or any corrections. The Version of Record will be available online at the publisher's website once the volume is published. Use of this Accepted Version is subject to the publisher's Accepted Manuscript terms of use
<https://www.springernature.com/gp/open-research/policies/accepted-manuscript-terms>

Density-Driven Emergent Division of Labor in Evolutionary Multi-Robot Systems

Paolo Leopardi^{1,2}[0009–0005–7064–9344], Heiko Hamann^{1,2}[0000–0002–2458–8289],
Jonas Kuckling^{1,2}[0000–0003–2391–2275], and
Tanja Katharina Kaiser³[0000–0002–1700–5508]

¹ Centre for the Advanced Study of Collective Behaviour, University of Konstanz,
Konstanz, Germany

² Department of Computer and Information Science, University of Konstanz,
Konstanz, Germany

`{firstname.lastname}@uni-konstanz.de`

³ Department of Computer Science and Artificial Intelligence, University of
Technology Nuremberg, Nuremberg, Germany
`tanja.kaiser@utn.de`

Abstract. Task specialization can enable robot teams to work more efficiently. However, it usually requires substantial engineering effort to decompose a task, design subtask controllers, and assign roles. Natural collectives suggest that specialization can emerge through adaptive behavior as groups scale or density increases. In this paper, we study the emergence of task specialization in an evolutionary multi-robot setting. Our example task is object retrieval, which allows for two interdependent subtasks. We study a homogeneous multi-robot system and evolve artificial neural networks as robot controllers. Without assigning subtasks or rewarding specialization, we observe emergent task specialization. Our results show a density-dependent transition from predominantly generalist behavior at low densities to emergent task specialization at higher densities. Given that our robots have no memory, their initial positions act as an implicit task-allocation cue. Together, these results provide a model of how division of labor can emerge from density-dependent task ecology, embodiment, and interaction constraints, bridging biological and robotic perspectives on multi-agent organization.

Keywords: Evolutionary Multi-Robot Systems · Division of Labor · Emergent Task Specialization · Object Retrieval.

1 Introduction

Social animals and social insects exhibit efficient collective behaviors through self-organized division of labor [24]. Various factors can trigger the self-organized division of labor, such as colony size and task demand [8]. Division of labor can also be beneficial to multi-robot systems (MRSs), where allocating tasks among robots can improve overall efficiency, for example, by reducing physical interference through spatial separation [16]. Decomposing a task into subtasks may also

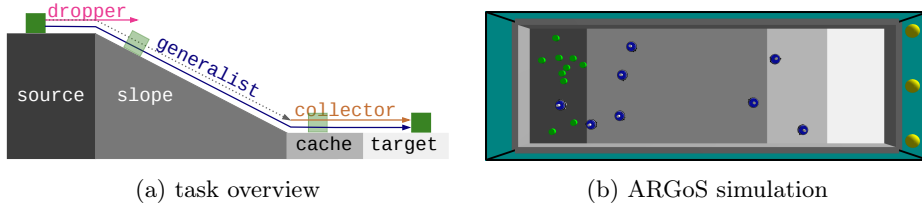


Fig. 1: Object retrieval task (a) and simulated arena in ARGoS (b) with blue robots and green objects. The task can be done by *generalist* robots or as two subtasks: *droppers* push objects down the slope into the cache, from where *collectors* take them to the target. Part (a) is adapted from [15].

reduce the complexity of robot controllers [12, 22]. Next to manually partitioning and implementing subtasks, task specialization in multi-robot systems can emerge during learning robot behaviors without being explicitly enforced [4, 6, 17]. In this work, we investigate how ecological structure, embodiment, and interaction constraints in multi-robot systems affect the emergence of task specialization in multi-robot systems, analogous to that observed in biological systems. In particular, we study density-dependent collective organization, that is, the effect of robot density on task partitioning and allocation, when evolving robot controllers under sparse reward signals for task completion.

As our benchmark, we use an object retrieval task inspired by the foraging of leafcutter ants [6, 10]. Objects must be transported from a source to a target via a slope and a cache, see Fig. 1a. Agents can use the slope to simplify their task, as objects can slide down into the cache. Through self-organized task partitioning into two subtasks, the travel distance required by individual agents can be reduced [24]. Object handover is indirect: one subgroup pushes objects down the slope into the cache, which acts as temporary storage, while another subgroup retrieves them from there. Agents do not need to coordinate with one another for object handovers and can operate asynchronously in different regions of the environment. Despite this decoupling, the subtasks remain interdependent and sequential, as the collecting agents can only operate if another agent has deposited an object in the cache beforehand. In robotics, similar two-stage workflows are common in warehousing and intralogistics [14].

We use a physics-based simulation to study how system size and density affect emergent task specialization. Our results show that higher robot densities promote evolved specialist behaviors, while task allocation depends on initial robot positions and is therefore vulnerable to failure.

2 Related Work

Division of labor describes the organization of collective work through task partitioning, in which subtasks are distributed among individuals or subgroups, po-

tentially leading to task specialization. It is widely studied in MRSs, with object-retrieval scenarios, such as foraging, commonly used as benchmarks [6,19,23].

Several studies have investigated environmental factors that promote division of labor and task specialization. According to the “economic-transport hypothesis” [24], it is promoted when environmental features can be exploited to increase efficiency. For example, object transport can be sped up by exploiting a slope in the environment [6]. In online distributed evolution of foraging with mutually exclusive resources in a flat environment [19], reproductive isolation (e.g., spatial separation) was identified as a primary factor in the optimization of division of labor. Secondary factors include large MRS sizes and selection mechanisms that promote exploration. Conversely, heterogeneous object distributions may negatively impact the emergence of specialization in this scenario [19]. In contrast to this work, we rely on a single resource type in an environment with a slope and offline evolution, that is, we optimize controllers before task execution. Hiraga and Ohkura [11] show that, in offline evolution, division of labor emerges in path formation tasks, especially under congestion (i.e., at high robot densities). The robots have memory through recurrent neural network controllers and can communicate via LEDs detectable by omnidirectional cameras. Contrary to this work, we use simple feedforward neural networks as robot controllers and have no inter-robot communication. We investigate whether task specialization can emerge in a minimal setting, without memory and only indirect robot coordination, with specialization driven by robot density.

Other works study the decision-making and whether task partitioning itself can emerge through evolution or learning. Pini *et al.* [22] study a threshold model [13] where robots decide individually whether to execute a subtask or the unpartitioned task based on cost estimates for the generalist strategy and the pre-partitioned subtasks. Ferrante *et al.* [6] evolved division of labor in a homogeneous robot swarm (i.e., same embodiment and control) *de novo* by rewarding group performance using grammatical evolution and predefined low-level behavioral primitives. The authors argue that task specialization is more likely to emerge when behaviors for subtasks are readily available as behavioral building blocks. In contrast, we optimize controllers without providing any low-level behavioral primitives. Furthermore, our robots can only push objects, rather than pick them up and release them, and they have no information about whether they are currently transporting an object. Van Diggelen *et al.* [4] combine the neuroevolution of specialist behaviors with a simple regulatory mechanism for task allocation. Genotypes encode controllers for different behaviors on a task, with subtasks not predefined, and group-level performance is rewarded during evolution. Task specialization is supported by phenotypic plasticity. Instead, we apply evolutionary optimization to artificial neural networks for object retrieval from scratch, without providing subtask-specific behaviors.

Brutschy *et al.* [3] examine the costs and benefits of specialization. They study task specialization in robots that search for tasks and learn to reduce processing time. Spatial effects, such as interference among robots, can increase the cost of specialization (search time) to the extent that it outweighs the benefits

(improved processing time). Instead of improving performance through learning, we reduce costs not directly related to task execution, such as transport distance.

3 Method

3.1 Task Description

We model a foraging scenario inspired by leafcutter ants harvesting leaves from trees [6]. The environment consists of four areas $\mathcal{A}$: source, slope, cache, and target, see Fig. 1a. M objects O_m , $m \in \{1, \dots, M\}$ are initially located in the source and can be retrieved at each time step t . Robots must retrieve as many objects as possible from the source and transport them to the target. When an object is deposited in the target, it disappears, and a new object is placed in the source area. Objects placed on the slope automatically slide down into the cache area, allowing robots to exploit the slope to accelerate object retrieval. We measure the team performance by the number of objects C_{target}^T collected over the time period T . Building on Balch [11], C_{target}^T is defined as follows:

$$C_{\text{target}}^T = \sum_{t=1}^T \sum_{m=1}^M H(O_m, t, \text{target}), \quad (1)$$

$$H(O_m, t, \mathcal{A}_i) = \begin{cases} 1, & \text{if object } O_m \text{ is in area } \mathcal{A}_i \text{ at time } t \\ 0, & \text{otherwise} \end{cases}, \quad (2)$$

where M is the number of objects in the environment at time t and $H(O_m, t, \mathcal{A}_i)$ indicates whether object O_m was in area $\mathcal{A}_i$ at time t . Each object is only counted once before disappearing. The foraging task can be solved using two strategies. In the *generalist* strategy, each robot independently transports objects from the source to the target. In the *specialist* strategy, the task is divided into two subtasks: *dropper* robots push objects from the source onto the slope, while *collectors* retrieve objects from the cache and deliver them to the target.

3.2 Experimental Setup

Arena. We use an arena of $2\text{ m} \times 6\text{ m}$ in the open-source simulator ARGoS [21], see Fig. 1b. The target, cache, and source are each 1 m long, and the slope is 3 m long. Three light sources are located above the target. The objects are cylinders with a radius of 0.05 m , a height of 0.08 m , and a mass of 0.1 kg . $M = 10$ objects are randomly placed in the source. Once an object is placed in the target, it is removed and replaced by a new object randomly placed in the source, ensuring unlimited target storage capacity. To simplify slope simulation, we model the arena as flat in ARGoS, but objects on the slope are automatically moved straight toward the cache at 10 m/s . Once at the bottom, the object is randomly repositioned along the cache length while retaining its width position. Robot speed on the slope is scaled by factors 0.2 for up and 1.5 for down.

Robots. We simulate homogeneous teams of foot-bots [2], a differential-drive mobile ground robot. Each robot has 24 light sensors, 24 proximity sensors, and 4 ground sensors. Light sensors enable robots to detect the light gradient toward the target area. Ground sensors allow to distinguish between the different environmental areas. We extend the platform with a 2D LiDAR providing 360° coverage. The LiDAR measurements are partitioned into eight sectors, each of which returns its minimum range value. By combining proximity and LiDAR readings, robots can distinguish objects detected only by proximity sensors from other robots or obstacles detected by both sensor types. We normalize all sensor values to $[0, 1]$ by their maximum range. Since robots can only push objects, they may lose contact with them when the objects slide down the slope. However, they can easily relocate the object in the cache by driving straight downhill. Robot speed is limited to $[0, 1]$ m/s.

Control Architecture. Each robot is controlled by a fully connected feedforward artificial neural network (ANN) with one hidden layer of eight neurons. All controllers in the same run share the same weights (i.e., homogeneous system). The ANN outputs the linear velocities (v_r, v_l) of the right/left wheel. The 60 controller inputs $s_1, \dots, s_{60}$ are defined as follows: $s_1, \dots, s_{24}$ are the light sensors readings, $s_{25}, \dots, s_{48}$ are the proximity sensors readings, $s_{49}, \dots, s_{52}$ are the ground sensors readings, and $s_{53}, \dots, s_{60}$ are the preprocessed LiDAR values.

3.3 Evolutionary Optimization Method

We optimize the ANN weights using evolutionary optimization [5], using DEAP [7] for the implementation and Hydra [25] for configuration management. Genomes encode ANN weights as floats, and initial values are uniformly sampled from $[-1, 1]$. Our evolutionary algorithm uses tournament selection with a tournament size of 5, no elitism, and one-point crossover with a 70 % probability. Each gene undergoes mutation with 20 % probability, using Gaussian mutation with zero mean and standard deviation of 0.3. The hyperparameters were set empirically. The evaluation time T_{eval} is 60 seconds, with 100 simulation ticks per second. The population size is always set to 60 and we do 2500 generations. Initial robot poses are uniformly randomized across the entire arena, and initial object positions within the source area, with both varying across generations.

Fitness F is defined as group fitness by the number of objects C_{target}^T collected by the MRS within evaluation time $T_{\text{eval}} = T$ (Eq. 1):

$$F = C_{\text{target}}^T = \sum_{t=1}^T \sum_{m=1}^M H(O_m, t, \text{target}) . \quad (3)$$

Following Nelson et al. [20], Eq. 3 is an aggregate fitness function, as it rewards task completion without taking into account how the task was accomplished. We evolve controllers for MRS size $S \in \{8, 14, 25, 50, 75, 100\}$ with a fixed number of $M = 10$ co-existing objects to investigate how MRS size influences the emergence of task specialization during the evolution. We chose $S \geq 8$

since the results of our previous work [15] indicate that a smaller MRS size will lead to generalists only. We perform ten independent evolutionary runs for each MRS size. The evolutionary runs are executed on a computing cluster using CPU nodes equipped with Intel Xeon Platinum 8358 with 64 cores each.

3.4 Post-evaluation Method

We post-evaluate the behavior of each evolved controller by running 100 independent evaluations to classify its behavior and measure performance. Each trial runs for a post-evaluation time $T_{\text{p-eval}}$ of 120 s, with 100 simulation ticks per second. Robots are uniformly randomly positioned throughout the entire arena, while objects are randomly placed in the source.

We use a threshold-based classifier to categorize each robot’s behavior during a trial based on the robot’s positions. We denote the time a robot spends at the top of the slope (i.e., source area) as t_t , and the time a robot spends at the bottom (i.e., cache/target area) as t_b . To account for a potential initialization phase before task execution, we only consider $t > 20$ s during classification. We classify the robot behavior r as

$$r = \begin{cases} \text{dropper} & \text{if } t_t > 0 \wedge t_b = 0 \\ \text{collector} & \text{if } t_t = 0 \wedge t_b > 0 \\ \text{generalist} & \text{otherwise} \end{cases} . \quad (4)$$

This means we classify a robot as a dropper if it spends all its time on the source and slope. Analogously, collectors spend their time only on slope, cache, and target. Robots that spend time in both the source and cache/target areas are classified as generalists. We evaluate controllers based on the number of objects collected C_{target}^T (see Eq. 1). To assess the contribution of each robot, we normalize C_{target}^T by the MRS size S , defining $\bar{C}_{\text{target}}^T = C_{\text{target}}^T / S$.

We assess whether the different proportion of generalists across MRS sizes S is statistically significant using the Mann-Whitney U test with p -values adjusted via the Holm-Bonferroni correction. The adjusted p -values for the pairwise test between MRS sizes S_1 and S_2 are denoted by p_{S_1, S_2} ¹.

4 Results

4.1 Evolutionary Optimization Results

For all MRS sizes except $S = 14$, the fitness curves converge within 2500 generations, see Fig. 2. For $S = 14$, the best-evolved individuals of the last generation are either generalists or task-specialists in roughly equal proportions, explaining the lack of fitness convergence due to their different performance levels (see Sec. 4.2). In the last generation, we find a mean best fitness F (Eq. 3) of 19.4

¹ Complete statistical results, additional plots, and videos are available in our supplementary material: <https://doi.org/10.5281/zenodo.20799667>.

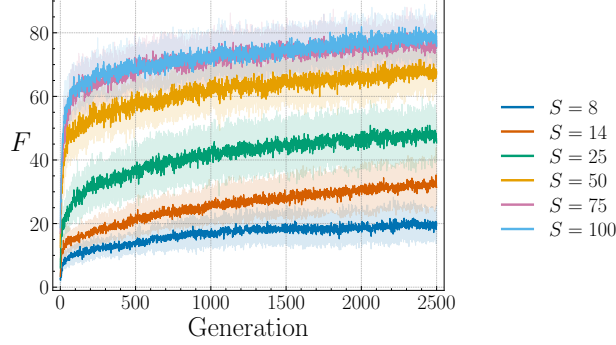


Fig. 2: Best fitness F (Eq. 3) over generations for 10 independent evolutionary runs per MRS size S (solid line: mean, shaded area: standard deviation).

($S = 8$), 31.4 ($S = 14$), 45.4 ($S = 25$), 67.7 ($S = 50$), 75.5 ($S = 75$) and 78.9 ($S = 100$). The best fitness increases with MRS size S , although with diminishing returns ($F/S = 2.43$ for $S = 8$, $F/S = 0.79$ for $S = 100$). The similar fitness curves for $S = 75$ and $S = 100$ suggest that robot density has become too high to yield further performance gains. For $S \geq 75$, physical interference among robots can thus outweigh the benefits of adding more robots. Robots on the slope may also temporarily block sliding objects, delaying their retrieval.

4.2 Post-evaluation Results

Behaviors. We find that our evolutionary approach produces both generalist and specialist strategies, see Sec. 3.1. Evolved generalists show a sawtooth motion pattern after a transient phase, see Fig. 3a. During the transient, robots orient themselves before following position-based motion rules. That is, robots use one side of the slope to move uphill and the other to move downhill. This is potentially due to two factors: (i) the potential loss of physical contact with an object while on the slope and (ii) the stateless robot controller (see Sec. 3.2). For example, a robot on the slope cannot infer from its sensor alone whether it has just pushed an object down the slope or whether it should move uphill to retrieve another object. Using opposite sides of the slope for uphill and downhill movement spatially encodes the two behavioral phases that result in consistent clockwise or counterclockwise robot trajectories through the arena. Task-specialist controllers direct each robot to its respective working area, where it operates exclusively as either a *dropper* at the source or a *collector* at the cache and target. As in the generalist strategy, specialist robots exhibit circular movement.¹ In addition, we observe hybrid behavior in which robots switch between the two subtasks during evaluation. As shown in Fig. 3c, robots cross the slope after performing one of the subtasks for some time and then switch to the opposite side of the arena.

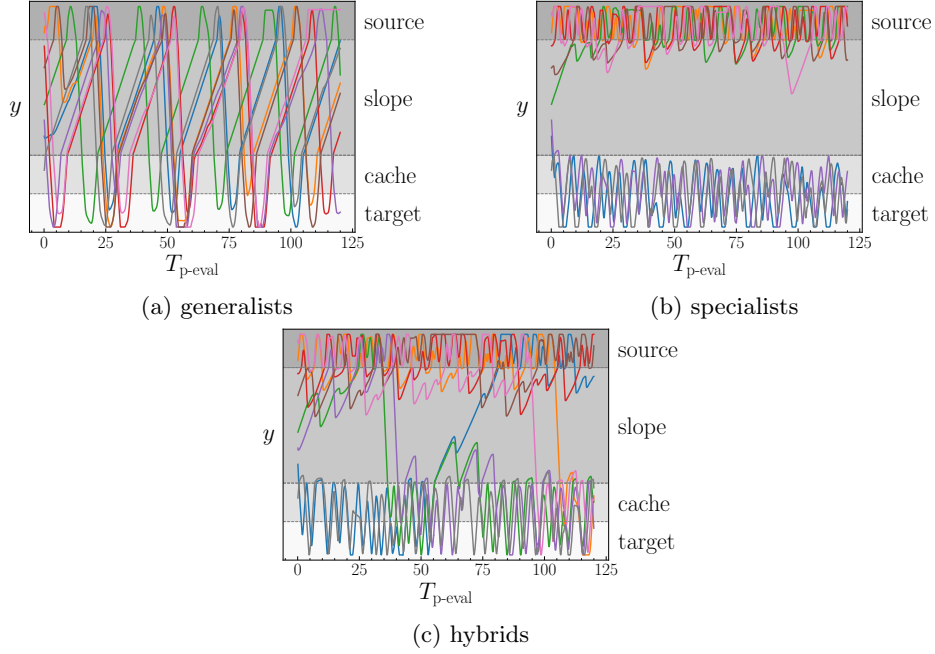


Fig. 3: Exemplary y -axis trajectories showing each robot’s current area for the generalist (a), specialist (b), and hybrid (c) strategies over $T_{p\text{-eval}} = 120$ seconds with $S = 8$ robots. Gray-shaded regions indicate the four areas, and each line represents one robot. Generalists move across the entire arena, specialists remain within their respective areas, and hybrid robots switch subtasks by moving from one side of the arena to the other.

Emergence of Specialization. Next, we investigate the effect of the robot density (i.e., MRS size S). As S increases, more task-specialist controllers evolve ($p_{8,14} = 0.135$, $p_{8,25} < 0.05$, $p_{14,25} < 0.05$), see Fig. 4.¹ From $S = 25$, all controllers lead to task specialization, and the allocation is approximately fifty-fifty. For $S \geq 75$, the high robot density increases interference between robots, potentially pushing robots outside their initial operating area and causing subtask switches. Our classification labels such robots as generalists, arguably due to conservative thresholds. We always observe a nonzero percentage of this hybrid robot behavior, so we do not classify any controllers as pure specialists.

We hypothesize that specialist robot controllers select their subtask based on their initial position (indicated by y_0). To test, we evaluate each evolved controller on a single robot initialized along a grid. The robot’s initial orientation is perpendicular to the direction of the slope to avoid bias toward either source or target. Robots with generalist controllers visit all areas of the arena independent of their initial position (see Fig. 5a). They spend most of their time on the slope due to reduced speed going uphill and the larger size of this area. In contrast,

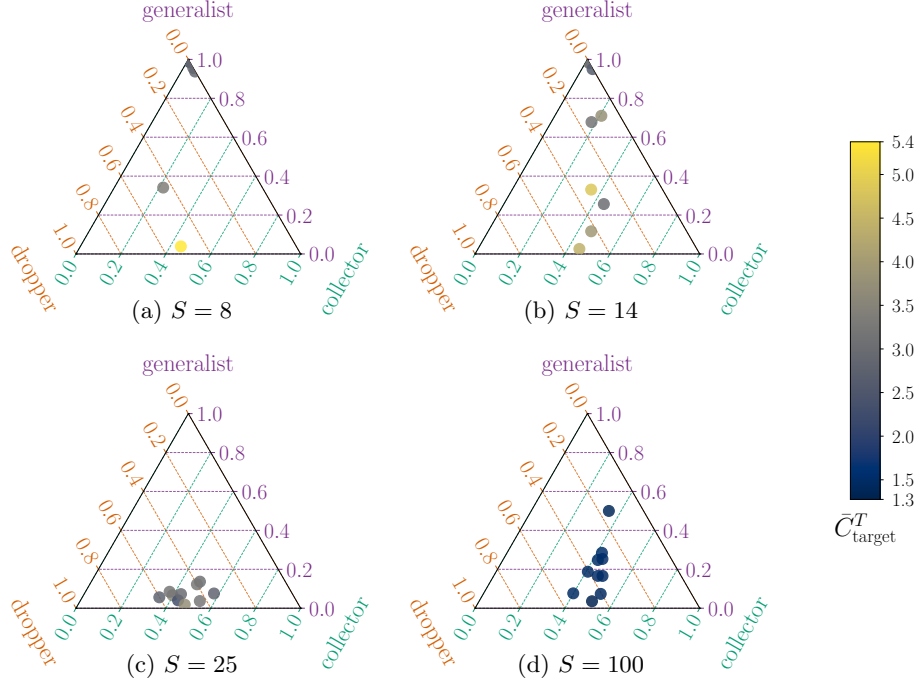


Fig. 4: Ternary plot for MRS sizes, each point represents the best-evolved controller of one evolutionary run, positioned according to its average role distribution: percentages of times spent as generalist, dropper, and collector are measured, and per-evaluation triplets are then averaged across 100 evaluations. $S = 25$ is also representative for $S \in \{50, 75\}$. Point color encodes the controller’s median fitness normalized by MRS size given by $\bar{C}_{\text{target}}^T$ (see Sec. 3.4).

robots with specialist controllers only operate on one side of the arena depending on their initial position (see Fig. 5b). When we initially position robots in the target, cache, or at the lower slope, they act as collectors. When they start on the slope or in the source, they act as a dropper. The exact boundary on the slope varies between controllers. Although a robot’s initial position causes the executed specialist behavior, sensor readings may vary and have additional influences. Especially for larger S , LiDAR and IR sensor values may differ significantly between single- and multi-robot experiments. Still, when evaluated at the evolved system size, robots specialize according to the described principle.

Individual Robot Performance. Finally, we evaluate the evolved controllers based on the number of collected objects per robot $\bar{C}_{\text{target}}^T$ (see Sec. 3.4). For $S = 8$ and $S = 14$, we find that specialized controllers outperform generalists (Spearman $\rho = -0.648$ and $\rho = -0.644$, respectively), see Figs. 4a-4b. From $S = 25$ (Figs. 4c-4d), only specialist controllers emerge, without notable per-

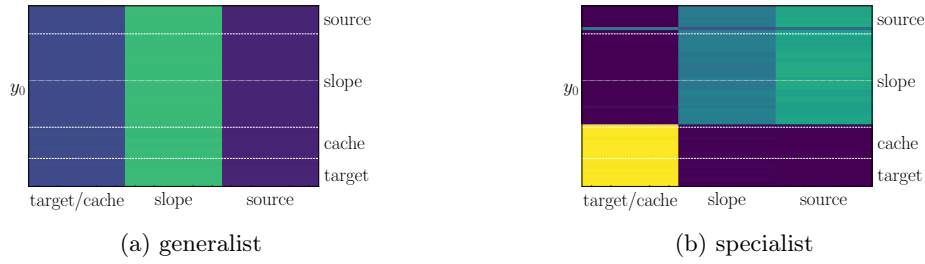


Fig. 5: Time spent per area (target or cache, slope, source) by a single robot initialized at a regular grid of positions over the entire arena for an exemplary generalist (a) and specialist (b) controller evolved with $S = 8$. Results are summed over all starting x positions and shown as a function of the initial y position y_0 ; brighter colors: robot spent more time.

formance differences. This indicates that, also for larger MRS sizes, task specialization remains the more efficient strategy. Across MRS sizes, we find that robots in sparser settings collect more objects. As S increases, individual efficiency decreases due to greater physical interference among the robots and to objects sliding down the slope at higher densities.

5 Discussion and Conclusion

In this work, we show that task specialization can emerge through evolution without being explicitly selected for. Robots are allocated to specialized roles according to their initial positions. This demonstrates that specialization can emerge in a MRS with homogeneous, memory-less controllers, without requiring heterogeneity [19], communication [11], or explicit awareness of subtasks [22, 64].

Our scenario fulfills the *economic-transport hypothesis* [24], that is, task specialization improves efficiency as specialized robots reduce travel distance by exploiting the slope. However, we observe that the emergence of task specialization is mainly driven by robot density and MRS size. At small MRS sizes, only a few runs yield specialized behavior, whereas at larger sizes, task specialization emerges consistently. We attribute this to sampling variability in the robots' initial positions. Robots will execute the subtask solely based on their initial location, regardless of how many robots are performing the other subtask. This emergent position-based task allocation has a major limitation: the task fails if all robots are initialized in proximity to one another and consequently choose to work on the same subtask. For large MRS sizes, robots are more likely to distribute across both sides of the slope during random initialization, with sufficient numbers on each side to cover both subtasks properly. In contrast, for small MRS sizes, unbalanced initializations are more likely, making the specialized behavior less predictable. As a systematic effect, the emergence of division of labor increases with increasing MRS size S . As we varied robot density via MRS size

rather than area, we cannot yet separate density from absolute MRS-size effects. We will test this by scaling area with robot number to keep density constant and assess whether the generalist–specialist tradeoffs observed for $S \leq 14$ persist.

Our simple threshold-based classifier strictly assigns dropper or collector roles. For example, a dropper that visits the cache shortly would be classified as a generalist. Hence, we overestimate the number of generalists. Still, for MRS sizes $S = 25$, $S = 50$, and $S = 75$, nearly all robots are classified as either dropper or collector. We also observe a hybrid behavior that combines elements of specialists and generalists. That is not explicitly captured by our classifier either. In future work, a more sophisticated classifier could provide more insights.

Future work should study the scalability aspects. Increasing MRS size from $S = 75$ to $S = 100$ did not improve performance due to interference among robots and between robots and objects on the slope. For larger MRS sizes, performance may decrease further due to interference on the slope [9], or an *avoid* role may emerge to keep robots off the slope [18]. To address both the limitations in scalability and task allocation, we will also consider more sophisticated control architectures that account for memory, communication, and better spatial awareness of the individual robots. In addition, we plan to compare the evolved controllers with hand-coded ones. This will inform us whether evolution discovers better actions and coordination mechanisms than human-designed systems.

Acknowledgments. PL, HH, and JK acknowledge support from DFG through Germany’s Excellence Strategy-EXC 2117-422037984 and Centre for the Advanced Study of Collective Behaviour (CASCb), University of Konstanz, Konstanz, Germany. JK acknowledges support from the Zukunftskolleg and the Carl-Zeiss-Foundation. The authors acknowledge support by the state of Baden-Württemberg through bwHPC.

Disclosure of Interests. The authors have no competing interests to declare.

References

1. Balch, T.: Taxonomies of multirobot task and reward. In: Balch, T., Parker, L.E. (eds.) *Robot Teams: From Diversity to Polymorphism*, pp. 23–35. AK Peters/CRC Press (2002)
2. Bonani, M., Longchamp, V., Magnenat, S., Rétornaz, P., Burnier, D., Roulet, G., Vaussard, F., Bleuler, H., Mondada, F.: The marXbot, a miniature mobile robot opening new perspectives for the collective-robotic research. In: *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*. IEEE (2010)
3. Brutschy, A., Tran, N.L., Baiboun, N., Frison, M., Pini, G., Roli, A., Dorigo, M., Birattari, M.: Costs and benefits of behavioral specialization. *Robotics and Autonomous Systems* **60**(11) (2012). <https://doi.org/10.1016/j.robot.2012.05.018>
4. van Diggelen, F., De Carlo, M., Cambier, N., Ferrante, E., Eiben, G.: Emergence of specialised collective behaviors in evolving heterogeneous swarms. In: *Int. Conf. on Parallel Problem Solving from Nature (PPSN)*. Springer (2024)
5. Eiben, A.E., Smith, J.E.: *Introduction to evolutionary computing*. Springer (2015)
6. Ferrante, E., Turgut, A.E., Duéñez-Guzmán, E., Dorigo, M., Wenseleers, T.: Evolution of self-organized task specialization in robot swarms. *PLOS Computational Biology* **11**(8) (2015)

7. Fortin, F.A., De Rainville, F.M., Gardner, M.A., Parizeau, M., Gagné, C.: DEAP: Evolutionary algorithms made easy. *J. of Machine Learning Research* **13** (2012)
8. Gautrais, J., Theraulaz, G., Deneubourg, J.L., Anderson, C.: Emergent polyethism as a consequence of increased colony size in insect societies. *Journal of Theoretical Biology* **215**(3), 363–373 (2002). <https://doi.org/10.1006/jtbi.2001.2506>
9. Hamann, H., Reina, A.: Scalability in Computing and Robotics. *IEEE Transactions on Computers* pp. 1453–1465 (2022). <https://doi.org/10.1109/TC.2021.3089044>
10. Hart, A.G., Anderson, C., Ratnieks, F.L.: Task partitioning in leafcutting ants. *Acta ethologica* **5**(1) (2002)
11. Hiraga, M., Ohkura, K.: Evolutionary emergence of path formation with autonomous specialization in a robotic swarm. In: *Proceedings of the 2018 Conference on Artificial Life*. pp. 526–527. MIT Press (2018)
12. Khamis, A., Hussein, A., Elmogy, A.: Multi-robot task allocation: A review of the state-of-the-art. *Cooperative robots and sensor networks* (2015)
13. Krieger, M.J., Billeter, J.B.: The call of duty: Self-organised task allocation in a population of up to twelve mobile robots. *Robotics and Autonomous Systems* **30**(1-2) (2000)
14. Leet, C., Oh, C., Lora, M., Koenig, S., Nuzzo, P.: Task assignment, scheduling, and motion planning for automated warehouses for million product workloads. In: *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*. IEEE (2023)
15. Leopardi, P., Hamann, H., Kuckling, J., Kaiser, T.K.: The evaluation cost of task specialization in evolutionary multi-robot systems. In: *Proc. of the Genetic and Evolutionary Computation Conference Companion*. GECCO, ACM (2026). <https://doi.org/10.1145/3795101.3805286>
16. Lerman, K., Galstyan, A.: Mathematical model of foraging in a group of robots: Effect of interference. *Autonomous robots* **13**(2) (2002)
17. Li, L., Martinoli, A., Abu-Mostafa, Y.S.: *Emergent specialization in swarm systems*. In: *Int. Conf. on Intell. Data Eng. and Automated Learning*. Springer (2002)
18. Mayya, S., Pierpaoli, P., Egerstedt, M.: Voluntary retreat for decentralized interference reduction in robot swarms. In: *Int. Conf. on Robotics and Automation (ICRA)*. pp. 9667–9673. IEEE (2019)
19. Montanier, J.M., Carrignon, S., Bredeche, N.: Behavioral specialization in embodied evolutionary robotics: Why so difficult? *Frontiers in Robotics and AI* **3** (2016)
20. Nelson, A.L., Barlow, G.J., Doitsidis, L.: Fitness functions in evolutionary robotics: A survey and analysis. *Robotics and Autonomous Systems* **57**(4) (2009)
21. Pinciroli, C., Trianni, V., O’Grady, R., Pini, G., Brutschy, A., Brambilla, M., et al.: ARGoS: a modular, parallel, multi-engine simulator for multi-robot systems. *Swarm intelligence* **6** (2012). <https://doi.org/10.1007/s11721-012-0072-5>
22. Pini, G., Brutschy, A., Frison, M., Roli, A., Dorigo, M., Birattari, M.: Task partitioning in swarms of robots: An adaptive method for strategy selection. *Swarm Intelligence* **5**(3) (2011)
23. Pini, G., Brutschy, A., Scheidler, A., Dorigo, M., Birattari, M.: Task partitioning in a robot swarm: Object retrieval as a sequence of subtasks with direct object transfer. *Artificial Life* **20**(3) (2014). https://doi.org/10.1162/ARTL_a_00132
24. Röschard, J., Roces, F.: Cutters, carriers and transport chains: distance-dependent foraging strategies in the grass-cutting ant *Atta vollenweideri*. *Insectes sociaux* **50**(3) (2003)
25. Yadan, O.: Hydra - a framework for elegantly configuring complex applications. Github (2019), <https://github.com/facebookresearch/hydra>