Benefit of Varying Navigation Strategies in Robot Teams

Seyed A. Mohaddesi[1(✉)], Mary Hegarty[2], Elizabeth R. Chrastil[1], and Jeffrey L. Krichmar[1]

[1] University of California Irvine, Irvine, CA 92697, USA
{smohadde,chrastil,jkrichma}@uci.edu
[2] University of California Santa Barbara, Santa Barbara, CA 93106, USA
hegarty@ucsb.edu

Abstract. Inspired by recent human studies, this paper investigates the benefits of employing varying navigation strategies in robot teams. We explore how mixed navigation strategies impact task completion time, environment exploration, and overall system effectiveness in multi-robot systems. Experiments were conducted in a simulated rectangular environment using Clearpath PR2 robots and evaluated different navigation strategies observed in humans: 1) Route (RT) knowledge where agents follow a predefined path, 2) Survey (SW) knowledge where agents take the shortest path while avoiding obstacles, 3) Mixed strategies with varying proportions, such as 40% RT and 60% SW (0.4RT 0.6SW) and 60% RT and 40% SW (0.6RT 0.4SW), and 4) An additional strategy where agents switch from RT to SW 10% of the time (0.9RT 0.1SW). While SW strategy is the most time-efficient, RT strategy covers more of the environment. Mixed strategies offer a balanced trade-off. These findings highlight the advantages of variability in navigation strategies, suggesting benefits in both biological and robotic populations. Additionally, we have observed that human participants in a similar study would start on a route, and then 10% of the time switch to survey. Therefore, we investigate a 90% Route 10% Survey (0.9RT 0.1SW) strategy for individual team members. While a pure Survey strategy is the most efficient regarding time taken and a pure Route strategy covers more of the environment, a mixture of strategies appears to be a beneficial tradeoff between time taken to complete a mission and area coverage. These results highlight the advantages of population variability, suggesting potential benefits in both biological and robotic populations.

Keywords: Navigation · Teams · Route · Survey · Multi-robot systems

This work was supported by the Air Force Office of Scientific Research (AFOSR) Contract No. FA9550-19-1- 0306, and by the National Science Foundation (NSF-FO award ID IIS-2024633.

O. Brock and J. Krichmar (Eds.): SAB 2024, LNAI 14993, pp. 63–77, 2025.
https://doi.org/10.1007/978-3-031-71533-4_5

1 Introduction

Multi-robot navigation plays a vital role in advancing robotic technology, providing a dynamic solution to complex tasks that exceed the capabilities of individual robots. Its significance lies in enhancing efficiency, collaboration, and adaptability across various domains, including manufacturing, search and rescue operations, and autonomous vehicles. Coordinated movement among multiple robots enables them to collectively navigate intricate environments, share information, and optimize routes, addressing challenges that a single robot might find overwhelming.

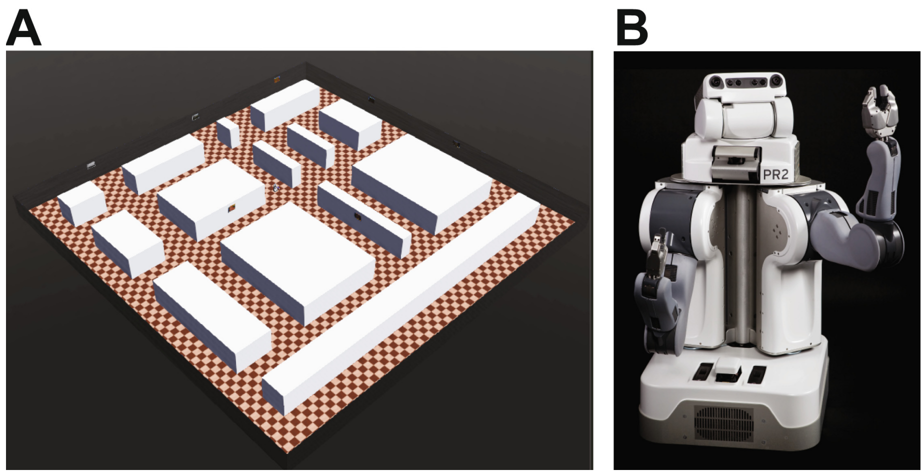

Fig. 1. (**A**) Overall view of our maze in the Webots environment (**B**) Clearpath's PR2 robot

Research suggests that people employ different types of knowledge to navigate [2]. Survey knowledge contains metric information that includes distances and directions between locations. This knowledge enables flexible path planning resulting in shortcuts or planned trajectories over never-experienced paths. In contrast, route knowledge consists of sequences of actions associated with places or decision points. Typically, the routes are fixed paths and inflexible.

Boone and colleagues [1] investigated the knowledge people use during navigation in the Dual Solutions Paradigm. In the DSP, participants follow a fixed loop around a virtual environment that has several landmarks along the way. After several laps, they are tested by placing participants at a landmark and telling them to go to another landmark. If they take the fixed loop, they are applying *route* knowledge. If they take a novel shortcut, they are applying *survey* knowledge. When told to take the shortest path, the proportion of participants applying survey knowledge increased. This suggests that many participants had survey knowledge, but they might find it easier to take a learned route.

In a metadata analysis, Krichmar and He showed that the variation observed in human navigation is both between and within subjects. This variability might be explained by taking the cost of traversing an environment into consideration [8]. They found that when told to find a goal, roughly 60% of participants used a route strategy and 40% used a survey strategy. However when told to take the shortest path to a goal, those proportions were reversed (40% route, 60% survey). Furthermore, they found that subjects starting on a route switched to a survey 10% of the time.

In the present paper, we simulate teams of robots to test whether there are advantages to using human-inspired strategies for navigating (Fig. 1). Our present work makes the following contributions:

- Simulating human variation has advantages for robot navigation, and possibly for planning algorithms in self-driving vehicles and robotic swarms.
- Incorporating a simple navigation strategy inspired by human subjects, rather than finding an optimal solution, makes multi-agent systems easier to scale.
- A mix of route and survey strategies leads to a tradeoff between time to find goals and more exploration of an environment that may be advantageous for biological and robotic populations.
- In human studies, it is technically challenging to monitor multiple participants navigating at the same time. Simulating a human-sized environment with navigating robots can overcome this limitation.

To better assess how these findings transfer to the real-world, we used a physical robot simulation with human sized robots and local sensing. In the following sections, background and methods are described in further detail.

2 Related Work

Several studies have explored the benefits of varying navigation strategies in teams of robots, addressing various aspects of multi-agent path planning and cooperative behavior. Most of the related work falls into two categories: 1) Multi-Agent Path Finding (MAPF), and 2) Heterogeneous swarm navigation.

2.1 Multi-Agent Path Finding (MAPF)

In MAPF, the goal is to plan collision-free paths for many agents to reduce mission duration and maximize team productivity. An example is the Conflict Based Search (CBS) algorithm which addresses the challenge of finding optimal paths in multi-agent scenarios by considering conflicts among agents and employing efficient exploration techniques [19]. Another example is the branch-and-cut-and-price (BCP) algorithm that incorporates a shortest path pricing problem for finding paths for every agent independently and constraints for resolving conflicts [9,10]. Both BCP and CBS are optimal but because of computational complexity, they don't scale well to a large number of agents [12]. Sub 1.5 MAPF algorithm on grids, optimizing time with a 1.5x longest-to-shortest path ratio constraint. This is a sub-optimal solution but it scales better [5]. Other algorithms scale well, by applying simple movement rules but are far from optimal [17,21]. MAPF-LNS is a hybrid approach that first creates a fast planning solution and then optimizes with a heuristic called large neighborhood search [13]. In the above cases, the solution is applied once for all agents. But a more realistic situation is a warehouse in which robots need to plan efficient, collision-free paths for long duration. The Rolling-Horizon Collision Resolution (RHCR) approach addressed the issue of lifelong multi-agent pathfinding in large-scale warehouses by applying their MAPF algorithm over different window periods [13].

This paper acknowledges the inherent limitations often found in human decision-making processes, in contrast to an emphasis on optimal solutions. While conflict resolution among robots remains a necessary aspect, it does not stand as the central objective within this paper's scope. However, future versions could take advantage of conflict resolution policies in the work discussed above. Unlike most MAPF algorithms, a centralized planner is absent in this work. Instead, each robot independently charts its path, driven by individual objectives, and limited knowledge of the other robot's state and intention.

2.2 Heterogeneous Swarm Navigation

Swarm robotics typically assumes that agents can communicate or interact with others in the vicinity. Heterogeneous teams of agents have been used to solve a wide range of problems [4,6,20]. In a heterogeneous robotic swarm, certain tasks can be solved efficiently through cooperation and functional specialization [3]. The agents or robots have different shapes or capabilities (e.g., different sensors or different locomotion). Unlike MAPF, the planning is decentralized. In a typical navigating swarm, there might be leaders and followers and the task is to find a goal while circumventing obstacles and preventing collisions [14,18]. Because of the locality requirement, the swarm often resembles a flock of birds.

Although there are similarities to our approach, these swarms by definition stay together. Furthermore, the majority of the prior work in swarm navigation assumes a 'bird's eye' view, rather than local sensing that would be required in many field operations. We are interested in heterogeneous foraging, where the robots are independent, decentralized and use different strategies.

3 Methods

3.1 Path Planning Algorithms

We simulated the Dual Solutions Paradigm (DSP) developed to study human behavior with teams of 1, 3, and 5 robots navigating the environment given in Fig. 2 which had 14 landmarks. To simulate survey knowledge, any path planner that finds the shortest path between a start and destination would be sufficient [11]. The present paper uses a spiking wavefront propagation algorithm, which has been described in [7], to calculate the survey paths. The spikewave algorithm takes into consideration the cost of traversal, which is encoded as a

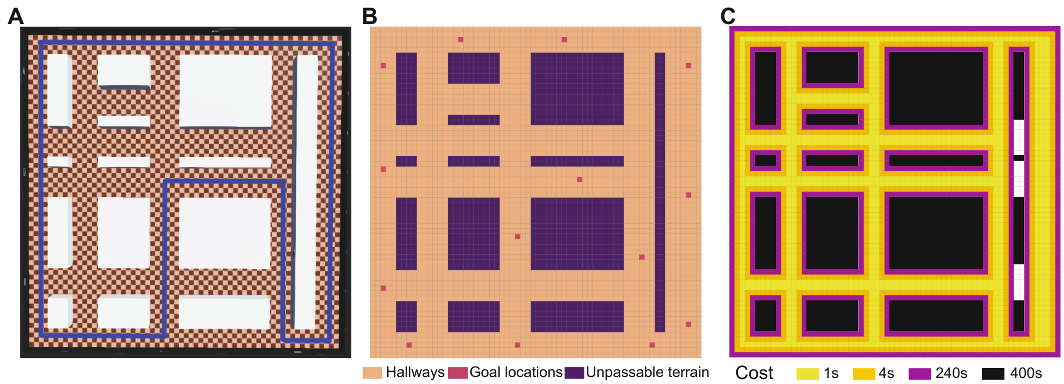

Fig. 2. (**A**) A bird's eye view of the map, The blue line indicates the specified route (**B**) Grid map of the environment. Purple cells show impassable terrain locations, goal locations are indicated in red, and beige cells indicate the floor of the aisles. (**C**) Heatmap depicting traversal delay costs measured in seconds for each cell on the map. The delay increases as cells approach the walls. This map serves as a universal guide utilized by all robots.

propagation delay. Difficult to traverse regions have long delays and impassable areas have very long delays (Fig. 2(c)). It is important to note that every path generated by spikewave is a sequence of straight line segments, meaning that there are no curves in generated paths. The yaw needed for the robot to face towards and move forward along the line segment is then determined. In order to simulate route knowledge, the robots followed a fixed path that took them past all the goal locations (Fig. 2(a)). Specifically, route knowledge was simulated with spikewave by giving the robot closely spaced waypoints to ensure it stayed on the specified route. As described below, the robots could either use route knowledge, survey knowledge or a mixture of these to find the goal locations.

3.2 Simulation Environment

In this experiment, a 3D simulated environment was created using Webots [15], an open-source robot simulator. The environment contains a square-shaped maze, which is made of 4096(64*64) cells each representing a square meter, as shown in Fig. 1(a). Up to 5 simulated PR2 robots, provided in the Webots environment, were used to explore the maze (Fig. 1(b)). The PR2 robot is a mobile robot with advanced sensors and software that can perform various tasks such as navigation and perception with high accuracy and precision. Its unique design features a mobile base with two wheels and a caster and a robotic arm with seven degrees of freedom. The maze size was scaled such that the robot fit within a cell.

3.3 Experimental Control and Design

The experimental design followed the DSP [1], but with multiple agents simultaneously navigating the environment. Based on the team's size, each robot is assigned a task involving a subset of randomly designated goals chosen randomly from the 14 landmarks (Fig. 2(b)). These goals are distributed among the robot team to ensure that each goal is visited by at least one robot.

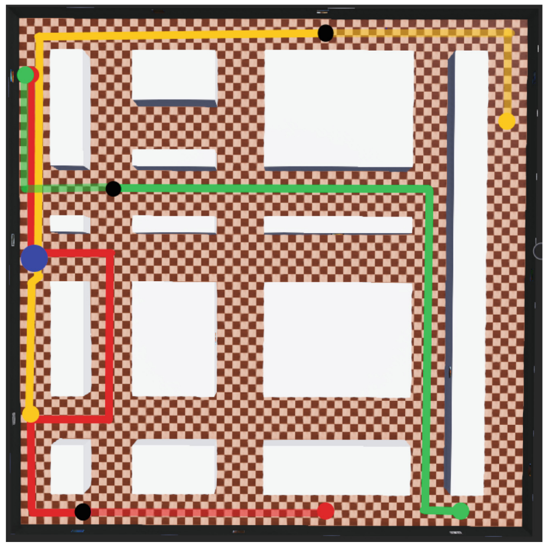

Fig. 3. Example of 3 Robots navigating through the environment, Black spots indicate the current location of each robot. Bold colored lines show the path each robot has taken till now, and pale colored lines indicate the path planned. The blue circle is the conflict spot between the yellow and red robots (Color figure online)

To conduct the simulation, several challenges need to be addressed. The first was to develop a subroutine that could handle movement between targets. This controller had to be capable of calculating a route between two targets that were free of obstructions and could be traversed by the robots. For the route strategy, an unobstructed path was provided to the robot as in the Dual Solution Paradigm used in human studies [1]. For the survey strategy, the spikewave algorithm uses the map with traversal delays to find the optimal path (Fig. 2(c)) as in [7]. An obstacle or environment boundary would have a long delay that would slow the wave propagation, and a traversable path would have a short delay that results in the wave finding a short, obstacle-free path.

Due to the requirement of simulating physical robots in the present work, several challenges needed to be addressed. One challenge is that sometimes agents come so close to colliding with a wall. The PR2's base Lidar, with its scanning range of 270°C, was used to measure the distance to the wall. If this length is less than 60cm, a subroutine is triggered to take the robot back to its original path. The spikewave algorithm then computes a new path to the goal. Another challenge was that a rule had to be devised and implemented to handle bottleneck conflicts and determine which robot has the right of way. Figure 3 is an example of the paths of 3 robots. For the purposes of this experiment, an ID was assigned to each robot (varying from 1 to 5). Two robots are in conflict when the distance between them is less than 6 m. When conflicts arise (Fig. 4(a)), the robot with the higher ID pauses, allowing the other robot to move a distance of at least 12 m away. This distance was empirically determined to effectively reduce the potential for subsequent collisions involving the PR2 robots. In cases where our defined route strategy is applicable, prohibiting rerouting, conflict resolution favors the robot that holds a positional lead on the route. Finally, since the problem of navigation strategies is similar to real-life scenarios, a single management unit could not be used to handle all conflicts between the robots. Instead, all decisions to control the disputes had to be either through predetermined rules or message passing among the agents. When a collision occurs, a signal is sent to all other robots within 25 m of the collision point. The robots that receive this signal increase the delay of the cells around the collision point on the delay matrix. The delay is increased by 500 s for a 3×3 square block around the collision point. Each agent then runs the spikewave algorithm to reroute to the temporary goal using the new matrix. This helps robots avoid the collision point and prevents a third robot from interfering with two conflicting robots (Fig. 4(b)).

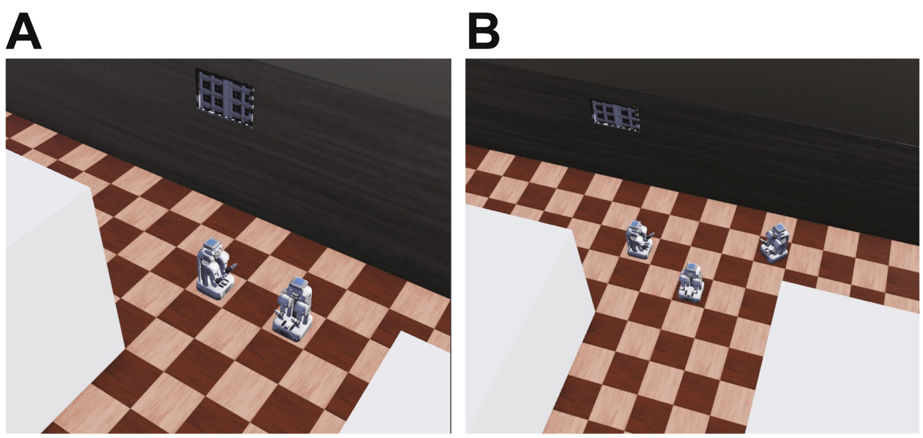

Fig. 4. Types of robot collisions (**A**) Collision of two robots. (**B**) Collision of three robots.

In order to automate these routines we designed a standalone controller with real-time path planning and online conflict resolution for all the robots. The pseudo-code that demonstrates how this controller works is provided (see Supplementary Algorithm 1). All agents used the same C++ implementation of this pseudo-code in their controllers.

To evaluate and compare different navigation strategies, we used these metrics:

- Time taken in seconds (s) till all goals have been visited.
- Total number of collisions between robots.
- Overall area occupied by all robots as a percentage of the entire environment (%).

– Overall intersection of the occupied area between robots as a percentage of the entire environment (%).

We ran 5 trials with 1, 3 or 5 robots in the environment using these strategies:

– Route (RT): Familiar route strategy. All the paths to the targets have to be aligned to a pre-specified route.
– Survey (SW): The shortest, obstacle-free path to the target is calculated using the spikewave (SW) algorithm.
– Mixed (0.9RT 0.1SW): Agents start on a route and 10% of the time the paths they switch to the spikewave algorithm at the halfway point.
– Mixed (0.6RT 0.4SW): 40% of the time the paths to the targets are calculated using the spikewave algorithm and 40% of the time it has to go along the route.
– Mixed (0.4RT 0.6SW): 60% of the time (i.e., 3 of the 5 trials) the paths to the targets are calculated using spikewave algorithm and 40% of the time (i.e., 2 out of the 5 trials) it has to go along the route.

The 3 mixed strategies represent variability observed within and between individuals. The percentages for the mixed strategies were derived from a DSP meta-analysis [8] that showed when asked to go to a goal participants roughly used SW 40% and RT 60% and when told to take the shortest path participants roughly used SW 60% and RT 40%. The (0.9RT 0.1SW) strategy was derived from the observation that when starting on a route, 10% of the time participants switch to a survey strategy halfway towards their goal.

4 Results

We present the results obtained from a comprehensive set of simulations that aimed to compare the performance of 5 distinct navigation strategies, namely RT, 0.9RT 0.1SW, 0.6RT 0.4SW, 0.4RT 0.6SW, and SW. The experiments were conducted using different team sizes of 1, 3, and 5 robots, allowing us to investigate the impact of team size and strategies on navigation performance. To ensure the reliability of our findings, each combination of strategy and robot number was subjected to five independent trials. Various metrics were measured throughout the trials, encompassing the time taken, the number of collisions, the area covered (occupancy), and team intersection. The metrics obtained from the trials were then averaged across the five trials for each experimental condition.

Time taken, defined as the duration in seconds from the start of the simulation until the visit of the last remaining goal, served as a crucial performance indicator. The SW strategy had a significantly shorter time to complete the task across all tested robot numbers (Fig. 5(A)). Conversely, the RT strategy exhibited the longest average time. This suggests that robots employing the SW strategy were able to devise innovative shortcuts, enabling them to complete the task more swiftly. For the complete statistical comparisons, see Supplementary Table 1. Additionally, we observed that increasing the number of robots led to

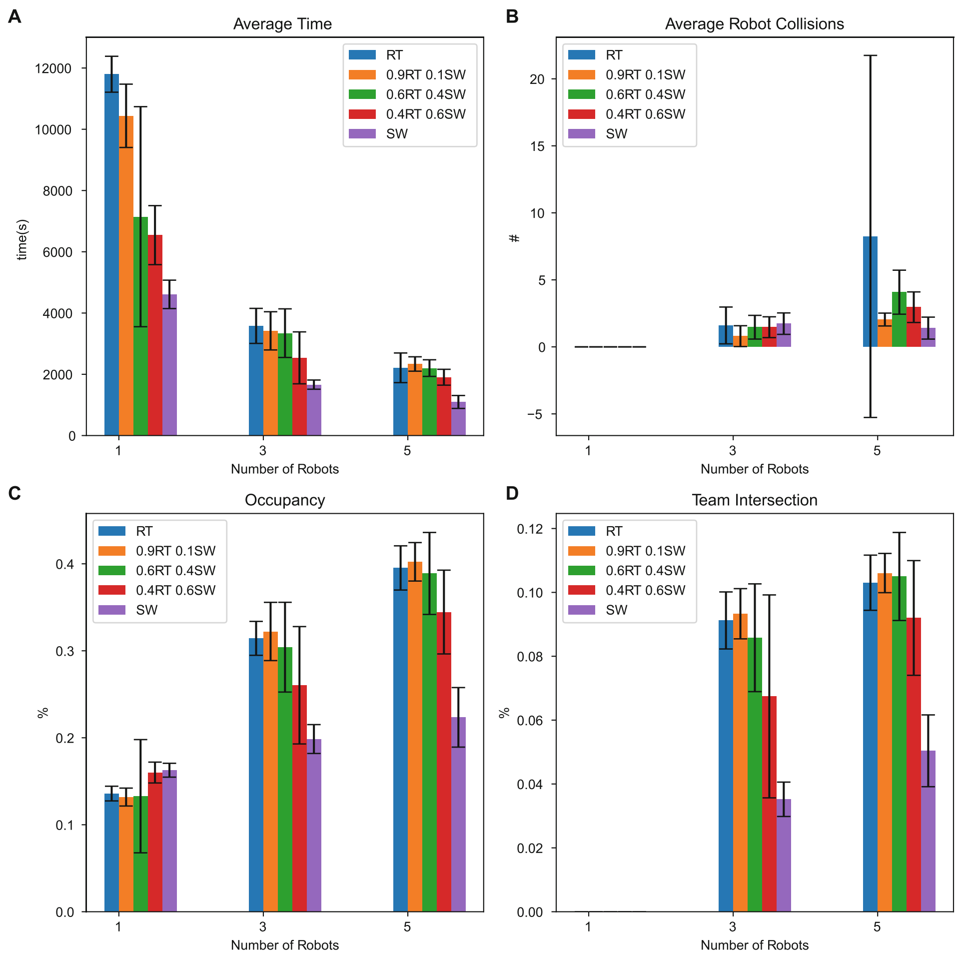

Fig. 5. Results of experiments for 1, 3, and 5 robots employing 5 different strategies, Route (RT), Survey (SW), and the mixed strategies (0.9RT 0.1SW, 0.6RT 0.4SW, 0.4RT 0.6SW). These values were measured and averaged over 5 trials. (**A**) Time to completion. Cumulative time taken to accomplish all goals (**B**) Robot collisions (**C**) Occupancy, which is the proportion of the environment covered by the robot teams (**D**) Intersection, which quantifies the percentage of cells visited by two or more robots

a significant reduction in the average time taken (see Supplementary Table 8 for statistical comparisons). This outcome can be attributed to the distribution of the goal locations among multiple robots, thereby enhancing efficiency.

We examined the metric of robot collisions, which denotes the total number of encounters between two robots. Notably, for a single robot scenario, this value is zero. The RT strategy exhibited the highest average number of robot collisions when tested with five robots (Fig. 5(B)). Although this may be due to an RT trial with 5 robots in which many collisions occurred, Increasing the number of robots tended to increase the number of collisions. It appeared that mixed strategies and the SW strategy had fewer collisions. However, it is hard to compare due to high variability from trial to trial (see Supplementary Tables 3 and 4 for statistical comparisons).

We analyzed the metric of occupancy, which reflects the extent of the map area covered by the robots. As anticipated, an increase in the number of robots resulted in significantly more area coverage (Fig. 5(C)) for all strategies except for SW when comparing teams of 3 to teams of 5 (see Supplementary Table 6). The RT strategy and mixed strategies covered more area than SW (Supplementary Table 5).

Finally, we evaluated the metric of team intersection, which quantifies the extent of intersected areas between robots on the map. Note that there is no intersection with only one robot. As the navigation strategies went from pure RT to mixed strategies to pure SW, the intersected areas between robots diminished significantly (Fig. 5(D)). Conversely, an increase in the number of robots resulted in a significant rise in this metric (see Supplementary Tables 7 and 8 for statistical comparisons).

Taken together, these results suggest that there may be a sweet spot where varying navigation strategies, like 0.9RT 0.1SW, 0.6RT 0.4SW and 0.4RT 0.6SW, can lead to shorter search times than a pure RT strategy and more coverage of the environment than a pure SW strategy. Moreover, increased intersections, as seen with RT and mixed strategies, may facilitate team communication, which is especially important in situations where lines of communication are down (e.g., disaster zones).

5 Discussion

The main finding of the paper was that although survey knowledge was the most time-efficient strategy and route-based strategies explored most regions of the environment, mixtures that varied strategies struck a balance between being efficient and covering more of the environment. As the number of robots increases tasks are completed faster, and the area covered expanded. The mixed strategies were a sweet spot of efficient navigation, area coverage, and collisions as the team size increased. Robot teams that had varied strategies completed the task of finding all the goal locations in less time than a pure route strategy while visiting more of the environment than a pure survey strategy. This heterogeneous strategy increases efficiency while gaining more knowledge compared to a homogeneous team strategy. In human populations, and potentially in robot teams, the choice of strategy depends on the context. In conditions where efficiency is critical, shortcuts that take more cognitive processing should be favored [8]. In contrast, a familiar route alleviates cognitive load and leads to more interactions with others.

Our results have significant implications for the design of heterogeneous swarms. Rather than varying the agent's sensing or locomotion capabilities [14,18], the agents used a mixture of strategies, which demonstrated clear population benefits. Unlike many multi-agent studies, the present study required to incorporate the physics of the environment with an accurate simulation of commercially available robots [15].These computationally intensive simulations limited team size. In the future, it would be of interest to test the benefits of varying strategies with larger, physical robot teams, which might be possible in an environment like the Robotarium [16].

In summary, the present results explain why human populations vary their navigation strategies and demonstrate that this variation is beneficial. By employing a physical robot simulation that used realistic local sensing and spacing of physical robots, we can better assess how the these results could transfer to the real world.

6 Supplementary Materials

6.1 Algorithms

Algorithm 1. Agent's controller pseudo-code

```
procedure CONTROLLER(self, robots, source, destination, strategy, costMap):
    while self → timeStep() do
        d[self] ← ∞
        newCM←costMap
        for agent: all agents except self do:
            d[agent] ← agent's distance from self
        end for
        [r,i] ← [min(d), argmin(d)]
        checkForObstructionSignal(&newCM)    ▷ checks if other robots have sent any obstruction signal, newCm was passed as reference to apply changes accordingly
        if r ≤ 6 then
            collision[self]++
            sendObstructionSignalTo(robots, 25)    ▷ Sends a signal to other robots within 25 meters of the conflict point to update their costMap subsequently
            if self → id ≤ i then
                newCM←newCM + block(3, robots[i])    ▷ block(size,robot) makes an all-zero 64*64 matrix except for a size * size square block centering at the location of robot with infinity value, used size=3 as an arbitrary value to avoid further collisions between these two robots
            else self→ waitFor(robot[i],12)    ▷ waitFor would stop this agent until its distance to robot[i] is at least 12 and then restores the original costMap for both robots
            end if
        end if
        path ← spikeWave (source, destination, strategy, newCM)    ▷ Shortest path from source to destination obeying the strategy
        navigate(path)    ▷ Navigate functions as a helper method that ensures traversal along a given path
    end while
end procedure
```

6.2 Statistical Comparisons

All statistical comparisons and p-values were generated using the two-sample Kolmogorov-Smirnov goodness-of-fit hypothesis test. Bonferroni corrections based on the number of comparisons were used to test for significance level (Table 2).

Table 1. Time to Completion. Comparing effects of strategy within same robot team. ** denotes $p < 0.01$ and * $p < 0.05$ after Bonferroni correction.

Strategy	1 Robot	3 Robots	5 Robots
RT vs. RT60%,SW40%	0.0038*	0.8899	0.9999
RT vs. RT40%,SW60%	0.0038*	0.0515	0.237
RT vs. SW	0.0038*	0.00001**	0.00001**
RT60%, SW40% vs. RT40%, SW60%	0.209	0.1359	0.237
RT60%, SW40% vs. SW	0.0361	0.00001**	0.00001**
RT40%, SW60% vs SW	0.0348	0.00001**	0.00001**

Table 2. Time to Completion. Comparing effects of strategy between different robot team sizes. ** denotes $p < 0.01$ and * $p < 0.05$ after Bonferroni correction.

Strategy	1 vs. Robots	1 vs. 5 Robots	3 vs. 5 Robots
RT	0.0003**	0.0001**	0.0006**
RT60%, SW40%	0.0066*	0.0037*	0.0098*
RT40%, SW60%	0.0003**	0.0001**	0.0006**
SW	0.0003**	0.0001**	0.00001**

Table 3. Collisions. Comparing effects of strategy within same robot team. ** denotes $p < 0.01$ and * $p < 0.05$ after Bonferroni correction.

Strategy	3 Robots	5 Robots
RT vs. RT60%, SW40%	0.9983	0.237
RT vs. RT40%, SW60%	0.8899	0.4141
RT vs. SW	0.9983	0.237
RT60%, SW40% vs. RT40%, SW60%	0.9999	0.4141
RT60%, SW40% vs. SW	0.9983	0.0038*
RT40%, SW60% vs SW	0.9983	0.0038*

Table 4. Collisions. Comparing effects of strategy between different robot team sizes. ** denotes $p < 0.01$ and * $p < 0.05$ after Bonferroni correction.

Strategy	3 vs. 5 Robots
RT	0.5898
RT60%, SW40%	0.0904
RT40%, SW60%	0.0363
SW	0.8634

Table 5. Occupancy. Comparing effects of strategy within same robot team. ** denotes $p < 0.01$ and * $p < 0.05$ after Bonferroni correction.

Strategy	1 Robot	3 Robots	5 Robots
RT vs. RT60%, SW40%	0.209	0.6974	0.9996
RT vs. RT40%, SW60%	0.0361	0.209	0.209
RT vs. SW	0.0038*	0.0038*	0.0038*
RT60%, SW40% vs. RT40%, SW60%	0.9996	0.6974	0.209
RT60%, SW40% vs. SW	0.6974	0.0038*	0.0038*
RT40%, SW60% vs SW	0.6974	0.209	0.0361

Table 6. Occupancy. Comparing effects of strategy between different robot team sizes. ** denotes $p < 0.01$ and * $p < 0.05$ after Bonferroni correction.

Strategy	1 vs. 3 Robots	1 vs. 5 Robots	3 vs. 5 Robots
RT	0.0038*	0.0038*	0.0038*
RT60%, SW40%	0.0038*	0.0038*	0.0361
RT40%, SW60%	0.0361	0.0038*	0.0361
SW	0.0038*	0.0361	0.209

Table 7. Intersection. Comparing effects of strategy within same robot team. ** denotes $p < 0.01$ and * $p < 0.05$ after Bonferroni correction.

Strategy	3 Robots	5 Robots
RT vs. RT60%, SW40%	0.1359	0.0258
RT vs. RT40%, SW60%	0.1359	0.0001**
RT vs. SW	0.00001**	0.00001**
RT60%, SW40% vs. RT40%, SW60%	0.1359	0.0001**
RT60%, SW40% vs. SW	0.00001**	0.00001**
RT40%, SW60% vs SW	0.0047*	0.00001**

Table 8. Intersection. Comparing effects of strategy between different robot team sizes. ** denotes $p < 0.01$ and * $p < 0.05$ after Bonferroni correction.

Strategy	3 vs. 5 Robots
RT	0.0012**
RT60%, SW40%	0.0012**
RT40%, SW60%	0.00001**
SW	0.00001**

References

1. Boone, A.P., Maghen, B., Hegarty, M.: Instructions matter: individual differences in navigation strategy and ability. Mem. Cognit. **47**(7), 1401–1414 (2019). https://doi.org/10.3758/s13421-019-00941-5
2. Chrastil, E.R., Warren, W.H.: Active and passive spatial learning in human navigation: acquisition of graph knowledge. J. Exp. Psychol. Learn. Mem. Cogn. **41**(4), 1162–1178 (2015). https://doi.org/10.1037/xlm0000082
3. Dorigo, M., et al.: Swarmanoid: a novel concept for the study of heterogeneous robotic swarms. IEEE Robot. Automation Magaz. **20**(4), 60–71 (2013). https://doi.org/10.1109/MRA.2013.2252996
4. Engelbrecht, A.P.: Heterogeneous particle swarm optimization. In: Dorigo, M., Birattari, M., Di Caro, G.A., Doursat, R., Engelbrecht, A.P., Floreano, D., Gambardella, L.M., Groß, R., Şahin, E., Sayama, H., Stützle, T. (eds.) ANTS 2010. LNCS, vol. 6234, pp. 191–202. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-15461-4_17
5. Guo, T., Yu, J.: Sub-1.5 time-optimal multi-robot path planning on grids in polynomial time. CoRR abs/2201.08976 (2022)
6. Hara, A., Shiraga, K., Takahama, T.: Heterogeneous particle swarm optimization including predator-prey relationship. In: The 6th International Conference on Soft Computing and Intelligent Systems, and The 13th International Symposium on Advanced Intelligence Systems, pp. 1368–1373 (2012). https://doi.org/10.1109/SCIS-ISIS.2012.6505194
7. Hwu, T., Wang, A.Y., Oros, N., Krichmar, J.L.: Adaptive robot path planning using a spiking neuron algorithm with axonal delays. IEEE Trans. Cognitive Dev. Syst. **10**(2), 126–137 (2018). https://doi.org/10.1109/Tcds.2017.2655539
8. Krichmar, J.L., He, C.: Importance of path planning variability: a simulation study. Top. Cogn. Sci. **15**(1), 139–162 (2023). https://doi.org/10.1111/tops.12568
9. Lam, E., Le Bodic, P.: New valid inequalities in branch-and-cut-and-price for multi-agent path finding. In: Proceedings of the International Conference on Automated Planning and Scheduling 30(1), pp. 184–192 (2020). https://doi.org/10.1609/icaps.v30i1.6660
10. Lam, E., Le Bodic, P., Harabor, D., Stuckey, P.J.: Branch-and-cut-and-price for multi-agent path finding. Comput. Oper. Res. **144**, 105809 (2022). https://doi.org/10.1016/j.cor.2022.105809
11. LaValle, S.M.: Motion planning part i: the essentials. IEEE Robot. Autom. Mag. **18**(1), 79–89 (2011). https://doi.org/10.1109/Mra.2011.940276

12. Li, J., Chen, Z., Harabor, D., Stuckey, P.J., Koenig, S.: Anytime multi-agent path finding via large neighborhood search. In: Zhou, Z.H. (ed.) Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21, pp. 4127–4135. International Joint Conferences on Artificial Intelligence Organization (8 2021). https://doi.org/10.24963/ijcai.2021/568, main Track
13. Li, J., Tinka, A., Kiesel, S., Durham, J.W., Kumar, T.K.S., Koenig, S.: Lifelong multi-agent path finding in large-scale warehouses. CoRR abs/2005.07371 (2020)
14. Maeda, R., Endo, T., Matsuno, F.: Decentralized navigation for heterogeneous swarm robots with limited field of view. IEEE Robot. Automation Lett. **2**(2), 904–911 (2017). https://doi.org/10.1109/LRA.2017.2654549
15. Michel, O.: Webots: professional mobile robot simulation. J. Adv. Robot. Syst. **1**(1), 39–42 (2004)
16. Pickem, D., et al.: The robotarium: a remotely accessible swarm robotics research testbed. In: 2017 IEEE International Conference on Robotics and Automation (ICRA), pp. 1699–1706 (2017). https://doi.org/10.1109/ICRA.2017.7989200
17. Sajid, Q., Luna, R., Bekris, K.E.: Multi-agent pathfinding with simultaneous execution of single-agent primitives. In: Fifth Annual Symposium on Combinatorial Search (2012)
18. Sano, Y., Endo, T., Shibuya, T., Matsuno, F.: Decentralized navigation and collision avoidance for robotic swarm with heterogeneous abilities. Adv. Robot. **37**(1–2), 25–36 (2023). https://doi.org/10.1080/01691864.2022.2117996
19. Sharon, G., Stern, R., Felner, A., Sturtevant, N.R.: Conflict-based search for optimal multi-agent pathfinding. Artif. Intell. **219**, 40–66 (2015). https://doi.org/10.1016/j.artint.2014.11.006
20. Valle, Y.d., Venayagamoorthy, G.K., Mohagheghi, S., Hernandez, J., Harley, R.G.: Particle swarm optimization: basic concepts, variants and applications in power systems. IEEE Trans. Evol. Comput. **12**(2), 171–195 (2008). https://doi.org/10.1109/TEVC.2007.896686
21. Wang, K.H.C., Botea, A.: Mapp: a scalable multi-agent path planning algorithm with tractability and completeness guarantees. J. Artif. Int. Res. **42**(1), 55–90 (2011)