Mobile stigmergic markers for navigation in a heterogeneous robotic swarm

F. Ducatelle, G. A. Di Caro, A. Förster, and L. M. Gambardella

Istituto Dalle Molle di Studi sull'Intelligenza Artificiale (IDSIA)
Galleria 2, 6928 Manno-Lugano, Switzerland
{frederick, gianni, alexander, luca}@idsia.ch *

Abstract. We study self-organized navigation in a heterogeneous robotic swarm consisting of two types of robots: small wheeled robots, called foot-bots, and flying robots that can attach to the ceiling, called eye-bots. The task of foot-bots is to navigate back and forth between a source and a target location. The eye-bots are placed in a chain on the ceiling, connecting source and target using infrared communication. Their task is to guide foot-bots, by giving local directional instructions. The problem we address is how the positions of eye-bots and the directional instructions they give can be adapted, so that they indicate a path that is efficient for foot-bot navigation, also in the presence of obstacles. We propose an approach of mutual adaptation, where foot-bots move according to eye-bot instructions, and eye-bots move according to observed foot-bot behavior. Our solution is inspired by pheromone based navigation of ants, as eye-bots serve as mobile stigmergic markers for foot-bot navigation. We evaluate the system's performance in a range of simulation experiments.

1 Introduction

We study how a heterogeneous robotic swarm composed of two sub-swarms can self-organize to solve a task. We are interested in mutual adaptation between sub-swarms: how can the sub-swarms adapt their behavior to each other, so that the swarm as a whole can solve the task. We focus on a navigation task, in which each sub-swarm plays a distinct role: the robots of one sub-swarm need to go back and forth between a source and a target location, while the robots of the other sub-swarm give guidance in this navigation task. For the first sub-swarm we use small wheeled robots, called foot-bots, and for the second flying robots that can attach to the ceiling, called eye-bots. Both these robots are under development in the project Swarmanoid (`http://www.swarmanoid.org`).

We deploy the robots in an indoor environment. We start from a situation where foot-bots are placed in the source location, and eye-bots are attached to

* This work was supported by the SWARMANOID project, funded by the Future and Emerging Technologies programme (IST-FET) of the European Commission under grant IST-022888. The information provided is the sole responsibility of the authors and does not reflect the Commission's opinion. The Commission is not responsible for any use made of data appearing in this publication.

the ceiling, forming a connected path between source and target (e.g., using the algorithm described in [1]). In this initial setup, eye-bots use infrared communication among them to derive the shortest path between the two locations. They locally give directional instructions to foot-bots passing below, so that these can follow this path. The main problem in this scenario is the presence of obstacles. If the environment contains obstacles (e.g., cupboards or sofas), the connected path formed by eye-bots near the ceiling communicating via an infrared device may pass over them. Such a path is difficult or impossible to follow for foot-bots. We investigate how eye-bots can adapt their positions and the directions they give in order to improve the navigability of the path they indicate.

We propose a distributed solution based on local adaptation between foot-bots and eye-bots. Foot-bots move from eye-bot to eye-bot following the directional instructions received from the eye-bots they pass. Eye-bots, in turn, adapt their position and their directional instructions based on the observation of foot-bots: they move to locations where they see a lot of foot-bots, and they adapt their instructions based on the directions where they see foot-bots come from. The former attracts them to areas that are navigable for foot-bots. The latter makes them indicate directions that are often followed by foot-bots. This way, eye-bots serve as mobile stigmergic markers for foot-bot navigation. In this sense, their role is similar to that of pheromone in ant-based navigation behavior. Later in this paper, we explain through examples from simulation experiments, how this process of mutual adaptation allows the heterogeneous swarm of eye-bots and foot-bots to find efficient paths in a wide range of different situations.

2 Robot characteristics and problem setup

We describe the features of the foot-bot and the eye-bot that are relevant for this work; further details about both robots are given in [2]. The foot-bot (Fig. 1(a)) moves on the ground. It has two cameras, one omnidirectional and one pointing up. It also has a rotating distance scanner. Foot-bots can communicate with each other and with eye-bots via visual signals, using the multi-color LED ring placed around their body and the LED beacon they have on top. Moreover, they can exchange wireless messages locally (up to 3 m) at low bandwidth using an infrared range and bearing (IrRB) system, which also gives them relative position information about each other. The eye-bot (Fig. 1(b)) is a flying robot, which can attach to the ceiling using a magnet (the design assumes ferromagnetic ceilings). It has a pan-and-tilt camera which it can point in any direction below. Like the foot-bot, it can communicate with visual signals using a multi-color LED ring, or with wireless messages using the IrRB system.

The eye-bots and foot-bots are placed in an arena like the one shown in Fig. 1(c). The task of foot-bots is to go back and forth between a source (top right in the figure) and a target location (bottom left in the figure) (e.g., to transport objects). The role of eye-bots is to support foot-bot navigation, by giving directional instructions to foot-bots that are within their visual range (the disk under selected eye-bots in the figure). All eye-bots are attached to

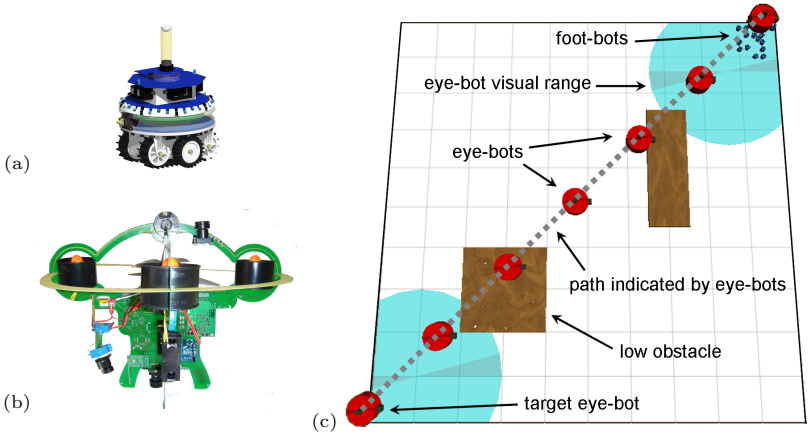


Fig. 1. (a) Foot-bot (CAD draw), (b) eye-bot (prototype), and (c) example scenario.

the ceiling. One is located near the source and one near the target. The others form a connected communication network between them, using the IrRB communication system (we study situations where the network is a single path, as shown in the figure, but also other connected formations could be used). Eye-bots derive the shortest path between source and target in the network (dotted line in the figure), and give directional instructions to foot-bots to follow this path. Foot-bots move from eye-bot to eye-bot following these instructions. The difficulty lies in dealing with obstacles. While the walls surrounding the arena reach from floor to ceiling, and can therefore easily be sensed by both foot-bots and eye-bots, other obstacles are lower (e.g., the two blocks in the middle in Fig. 1(c)), so that they block the way for foot-bots, but not for eye-bots or for IrRB communication between them. This means that the path set up between eye-bots may be difficult or impossible to follow for foot-bots on the ground. We investigate how the system can improve this path, by changing the positions of eye-bots and the directional instructions they give.

3 Related work

Our work is in the first place related to research on heterogeneous swarm robotics. Swarm robotics research has mainly focused on homogeneous systems. Nevertheless, there is some work using heterogeneous swarm robots for applications like flocking [3], where different but similar robots flock together like birds of distinct species might do, task allocation [4], where robots with different capabilities are assigned to different tasks, and recruitment [5], where robots of one type recruit robots of a different type. However, we know of no work where swarms of different robot types mutually adapt to jointly self-organize to solve a task.

In terms of the task to be solved, our work is related to research on self-organized foraging, where robots need to optimize a path to follow back and

forth between a source and a target [6–9]. All this work is inspired by pheromone guided foraging of ants in nature [10]. They use varying strategies to implement pheromone, such as, e.g., light projections [7], a map in a shared memory [8], or alcohol trails [6]. None of the existing work uses one swarm of robots to function as pheromone for another swarm, in the form of mobile stigmergic markers.

From an application point of view, we point out the relation with existing work on sensor network guided navigation [11–13]. They place communicating sensor nodes in the environment and let them cooperate to guide a single mobile robot to a target, similar to how eye-bots guide foot-bots. None of this work considers on-line adaptation of node positions to improve navigation, like we do. As a consequence, they need nodes to cover the full area in which robots are deployed. Moreover, they often use the communication links between nodes to find navigable paths for the mobile robot: they do not deal with the situation where obstacles block the way for the robot but not for node communication, which is central in this paper.

4 Self-organized path finding

4.1 General description

We start from a situation as shown in Fig. 1(c). In the beginning, eye-bots use network communication to derive the shortest route through the eye-bot network to the source and target locations. Using the relative position information given by the IrRB system, each eye-bot i derives from this routing information the directions θ_i^s towards the source and θ_i^t towards the target. These directions are broadcast locally to nearby foot-bots.

Foot-bots follow these directions, moving from eye-bot to eye-bot. When they encounter an obstacle, they use obstacle circumnavigation to go around it. They use light signals to give information to eye-bots (which observe them through their camera): to show where they are, which direction they are coming from, and whether they are going towards the target or the source.

Eye-bot actions consist in moving their position and changing their directions θ_i^s and θ_i^t (overriding the directions obtained from IrRB communication). Eye-bots move in the direction of areas where they observe foot-bots. This way, they are attracted to areas that are navigable for foot-bots and to paths that are often used by foot-bots. They also move away from nearby eye-bots, which makes them spread out and avoid collisions. Finally, they make reparatory moves when they loose network connectivity with source or target, which ensures that foot-bots can move between source and target while always staying within range of an eye-bot. Eye-bots adapt their directions θ_i^s and θ_i^t based on the direction where foot-bots going to respectively the target and the source are coming from: they assume that the direction where most foot-bots going to the target come from is a good indication of the direction to the source (and vice versa).

Through their adaptations to foot-bot behavior, eye-bots serve as mobile stigmergic markers for foot-bot navigation: they mark paths often followed by

foot-bots, and indicate them to other foot-bots. Their role is similar to that of pheromone in ant foraging. In Sect. 5, we show examples of how this works.

4.2 Giving directional instructions to foot-bots

Eye-bots give directional instructions to foot-bots using a combination of visual signals with LEDs and wireless communication with the IrRB system. Each eye-bot i switches on a red LED in front and a blue LED in the back, in order to show a reference direction θ_i^0. At regular intervals, it broadcasts θ_i^s and θ_i^t using the IrRB system. IrRB communication from an eye-bot i to foot-bots is focused in a cone below i, so that only foot-bots underneath i can receive its messages. In order to get directions, a foot-bot j moves under i. It uses its upward camera to define θ_i^0, and reads direction θ_i^s or θ_i^t (depending on whether j's goal is the source or the target) from the received wireless message. j interprets θ_i^s or θ_i^t relative to θ_i^0, in order to derive a new travel direction θ_j^n.

4.3 Foot-bot behavior

When a foot-bot j obtains a new direction θ_j^n from an eye-bot i, it turns into that direction, and moves forward for a default distance (enough to get out of the view of i), or until it arrives under a different eye-bot. If no other eye-bot was reached, j uses its upward camera to define the direction to the closest eye-bot, and moves there. If no eye-bot is seen, j starts a random movement: repeatedly make a random turn and move forward for a random distance. Given the nature of the task (to move from one location to another), foot-bots always avoid to locally retrace their movements.

When a foot-bot meets an obstacle while following θ_j^n or while going to an observed eye-bot, it executes an obstacle circumnavigation behavior. This is based on its distance scanner. The foot-bot moves parallel to the obstacle, for as long as it observes the direction it wanted to go in as blocked. Due to noise on the distance scanner and interference with other robots, obstacle circumnavigation has limited reliability.

Foot-bots use light signals to give feedback to eye-bots. A foot-bot j simultaneously switches on its LED beacon on top and one LED in front, in order to show its location and the direction it is coming from, θ_j^f. The color of the front LED indicates whether it is going towards the source or the target. In some cases, j can switch off its front led. This way, eye-bots still see where it is, but not its direction θ_j^f. As a consequence, eye-bots cannot adapt θ_i^s or θ_i^t based on θ_j^f. A foot-bot does this whenever its movement direction is not representative for the general direction it is following from source to target: when it performs obstacle circumnavigation or random movement. The goal is to limit errors in the eye-bot directions θ_i^s and θ_i^t.

4.4 Updating eye-bot positions

Each eye-bot i adapts its position in three different ways. The first is in the direction of observed foot-bots (to indicate good feasible paths for foot-bots).

The second is away from other eye-bots (to avoid collisions). The third is in the direction of lost communication neighbors (to repair connectivity in the eye-bot network). The first two are based on observations accumulated over time. The third is a reactive behavior triggered by loss of network connectivity. The eye-bots indicating the source and target locations never move. In the following these three behaviors are described.

When an eye-bot i observes a foot-bot j, it uses its camera observation and altitude measurement to calculate the relative distance r_{ij} and angle α_{ij} to j in i's horizontal plane. We indicate by $\mathbf{u_{ij}} = (cos(\alpha_{ij}), sin(\alpha_{ij}))$ the unit vector in the direction of j with respect to i's frame of reference (given by its reference direction θ_i^0). Using $\mathbf{u_{ij}}$ and r_{ij}, eye-bot i updates a two-dimensional vector $\mathbf{p_i}$, which it uses to direct its movements. After observing j, $\mathbf{p_i}$ is updated:

$$\mathbf{p_i} = \begin{cases} \mathbf{p_i} + (1 - r_{ij})\mathbf{u_{ij}} & \text{if } \ r_{ij} < r^f, \\ \mathbf{p_i} + (1 - r^f)\mathbf{u_{ij}} & \text{otherwise.} \end{cases} \tag{1}$$

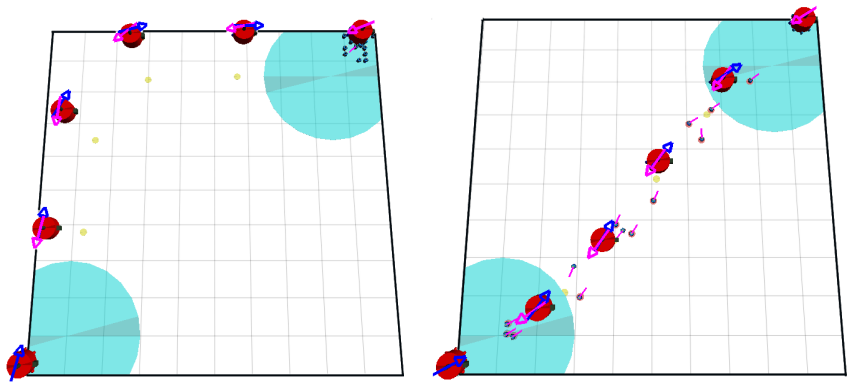
In this equation, $r^f \in [0, 1]$ is a distance threshold, which produces smaller updates for faraway foot-bots. Updating $\mathbf{p_i}$ for each foot-bot observation, eye-bot i calculates over time an aggregate of the directions in which it sees foot-bots. If foot-bots are observed more in one direction than in others, $\mathbf{p_i}$ grows in that direction. Once the magnitude of $\mathbf{p_i}$ reaches a threshold value c_p, $|\mathbf{p_i}| > c_p$, i makes a fixed small move in the direction indicated by $\mathbf{p_i}$. Then, $\mathbf{p_i}$ is re-initialized to $(0, 0)$. The reduced weight for faraway foot-bots in Eq. 1, based on the constant r^f, is meant to improve stability: faraway foot-bots passing by are observed in a given direction for longer than nearby foot-bots, which would make $\mathbf{p_i}$ grow too fast in their direction.

When i observes another eye-bot k nearby, it uses the IrRB system to derive the distance r_{ik} and angle α_{ik} to k. $\mathbf{u_{ik}} = (cos(\alpha_{ik}), sin(\alpha_{ik}))$ is i's unit vector in the direction of k. In this case, the same movement vector $\mathbf{p_i}$ is updated:

$$\mathbf{p_i} + e(r_{ik})\mathbf{u_{ik}}, \tag{2}$$

where $e(r_{ik})$ is a staircase function that serves to scale $\mathbf{u_{ik}}$ in different ways according to how far is eye-bot k. The closer k is, the larger is the scaling. This update makes $\mathbf{p_i}$ grow when two eye-bots get close to each other, so that eye-bots tend to spread out and avoid collisions.

Finally, eye-bots also make moves to repair network connectivity if lost due to relative eye-bot movements. Network connectivity is established as the result of running a network routing algorithm in the eye-bot network (in particular, we chose the Bellman-Ford algorithm [14]). Using the relative position information returned by the IrRB communication system, eye-bots derive the physical length of each communication link and use this to calculate the shortest path to source and target. The same information also allows to derive the direction to the next hop on each path. When Bellman-Ford routing fails to indicate a next hop, network connectivity is assumed to be lost. Then, eye-bots make a small move in the direction of the last known next hop on the lost path.



(a) (b)

Fig. 2. Open space experiments: (a) start positions, (b) convergence to a straight path

4.5 Updating eye-bot directions

Initial values for θ_i^s and θ_i^t are based on the directions to next hops as indicated by Bellman-Ford routing. Once an eye-bot starts observing foot-bots, it updates the values for θ_i^s and θ_i^t based on the directions where observed foot-bots come from. We explain the update of θ_i^s; for θ_i^t it is equivalent.

Internally, each eye-bot i represents the direction θ_i^s with a two-dimensional vector $\mathbf{v_i^s}$, which points in the direction of θ_i^s and initially has a size of 1. At discrete time intervals, each eye-bot i defines the set V_i^t of foot-bots that are in view of its camera and that are going towards the target (i.e., are coming from the source). For each foot-bot $j \in V_i^t$, it observes the direction it is coming from (the inverse of its movement direction), θ_j^f, based on the j's LED signals. It calculates $\mathbf{u_j^f} = (cos(d_j^f), sin(d_j^f))$, the unit vector in direction θ_j^f. Then, if $|V_i^t| > 0$, it updates $\mathbf{v_i^s}$ as in Eq. 3, and assigns the direction of $\mathbf{v_i^s}$ to θ_i^s.

$$\mathbf{v_i^s} = a\mathbf{v_i^s} + (1-a)\sum_{j \in V_i^t} \mathbf{u_j^f}, \quad \text{where } a \in]0,1[. \tag{3}$$

5 Experimental results

We investigate the behavior of the system through simulation tests using a range of different scenarios. All tests are done with the Swarmanoid simulator [15], which is developed as part of the Swarmanoid project. It contains precise models of the foot-bot and eye-bots robots. All experiments last 2000 seconds. We carry out 30 independent runs for each test.

5.1 Tests in an uncluttered environment: shortest path behavior

Looking at the behavior of our system in open space scenarios, we can understand how it finds efficient paths in open areas between subsequent obstacles. The

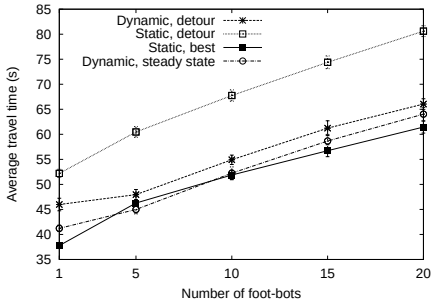


Fig. 3. Open space experiments: number of foot-bots vs. average travel time. Error bars show one standard deviation.

scenario is shown in Fig. 2(a): eye-bots start from an inefficient setup that makes a detour around the arena. Light and dark arrows above eye-bots indicate the directions they indicate towards respectively the target and source. The line segment above each foot-bot shows its movement direction.

Figure 2(b) shows a snapshot after 900 s: the eye-bots have formed an almost straight path between source and target. This is confirmed by numerical results in Fig. 3. We vary the number of foot-bots, and show the average time needed for a foot-bot to travel between source and target. Eye-bots start form the positions of Fig. 2(a) and move according to our algorithm. We measure the average travel time during the first 1000 s of simulation ("Dynamic, detour"), and between 2000 s and 3000 s, when the system has had time to converge onto a stable path ("Dynamic, steady state"). We compare to tests where eye-bots do not adapt their position or direction: we do tests where the eye-bots remain static in the positions of Fig. 2(a) ("Static detour"), and tests where they are placed in a straight line between source and target ("Static best"). In both cases they use IrRB communication to define directions. Results show that with our algorithm, foot-bot performance is close to that obtained over the straight path. In all cases, performance decreases with the number of foot-bots, due to congestion.

The ability to find straight paths relies on the tendency of an eye-bot to line up with neighbors that send foot-bots to it. An eye-bot that is not lined up with its neighbors observes foot-bots more in one direction than another, and moves in that direction. E.g., for the eye-bot in the top left in Fig. 2(a), foot-bots enter its field of view on the right (coming from the source) or at the bottom (coming from the target). Therefore, the eye-bot observes more foot-bots towards its bottom-right than towards its top-left half. Its movement vector $\mathbf{p_i}$ grows towards the bottom-right, and eventually the eye-bot moves in that direction. This process goes on continuously and lets eye-bots form straight lines.

5.2 Experiments in a cluttered environment

Here, we examine the setup of Fig. 1(c). In the beginning, eye-bots indicate a straight path between source and target. Foot-bots follow this path and use

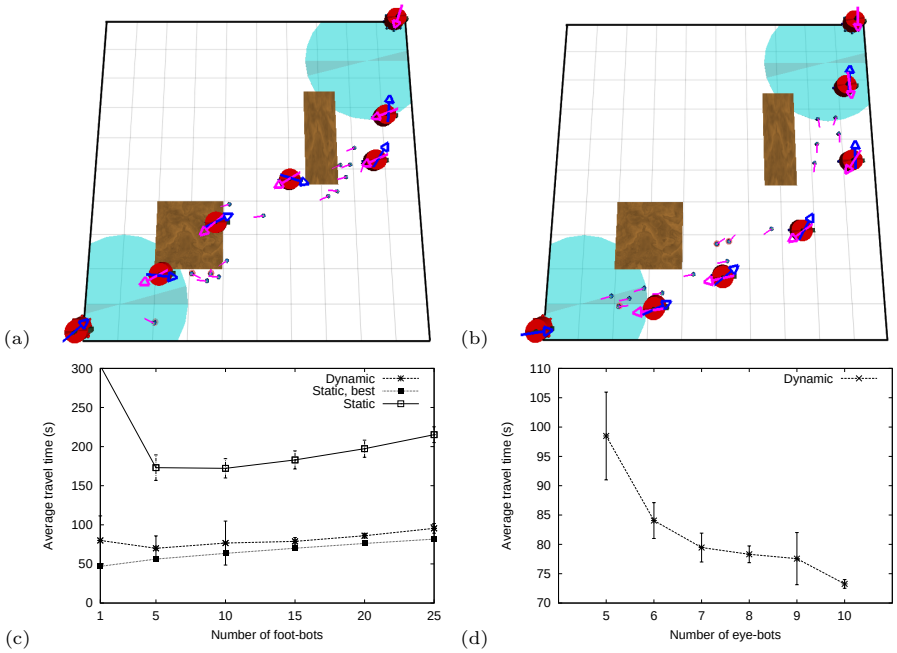
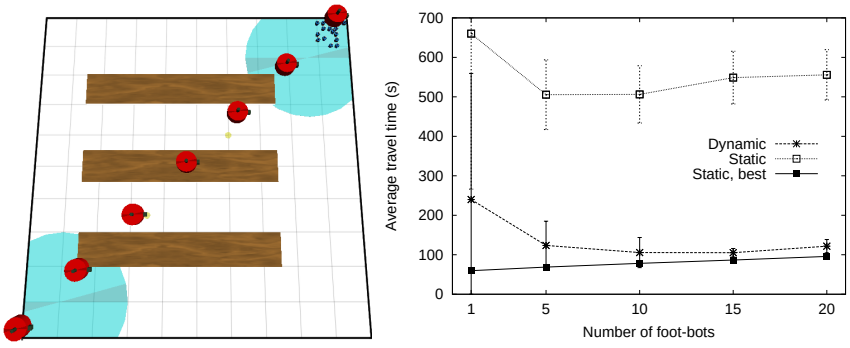


Fig. 4. Cluttered environment experiments: (a) foot-bots execute obstacle circumnavigation, (b) eye-bots find a feasible path, (c) number of foot-bots vs. travel time, (d) number of eye-bots vs. travel time. Error bars show one standard deviation.

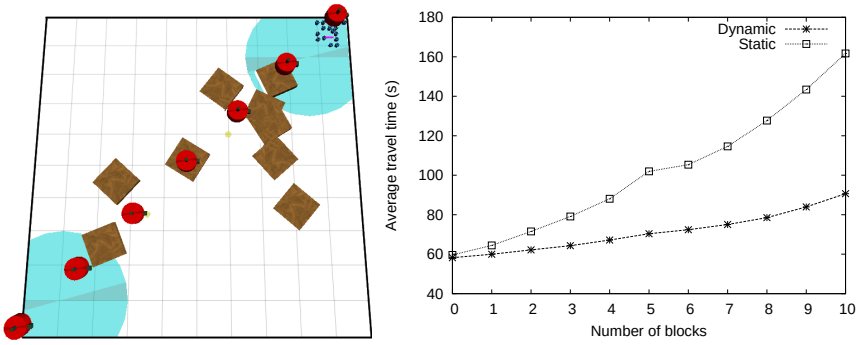
obstacle circumnavigation when their way is blocked. In Fig. 4(a), we show the situation after 122s. Foot-bots performing obstacle circumnavigation follow obstacle perimeters. Eye-bots move to locations where they observe foot-bots, so they tend to take place along obstacle edges. The directions indicated by eye-bots are adapted to the main movement directions of foot-bots, so they do not point towards obstacles. In the free space between obstacles, eye-bots form straight lines. Ultimately, a path of line segments connecting obstacle corners emerges. This is observed in Fig. 4(b), which shows a snapshot after 280s.

We measure the average time needed by foot-bots to travel between source and destination. In Fig. 4(c), we vary the number of foot-bots. We show results for tests where eye-bots use our adaptive behavior ("Dynamic"), tests where eye-bots remain static in the positions of Fig. 1(c) ("Static"), and tests where eye-bots remain static in a pre-defined path efficient for foot-bots ("Static, best"). The graph shows that our dynamic approach significantly improves foot-bot navigation efficiency compared to the initial placement, and obtains results close to those of the efficient path. For low numbers of foot-bots (1 to 10) results are more variable. This is because obstacle circumnavigation errors of single robots affect eye-bot behavior less when there are many other robots around. In Fig. 4(d), we vary the number of eye-bots, and show results for our adaptive



(a) (b)

Fig. 5. Three walls experiments: (a) initial setup, (b) number of foot-bots vs. travel time (error bars show one standard deviation)



(a) (b)

Fig. 6. Random obstacles experiments: (a) example setup with 10 blocks, (b) number of blocks vs. run delay

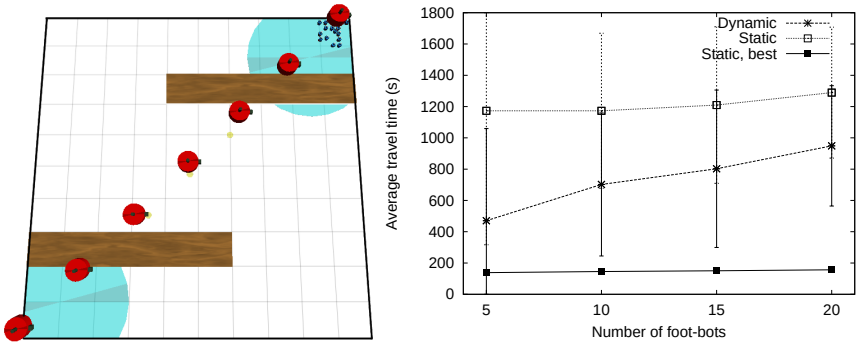
approach (using 15 foot-bots). The graph shows how a higher number of eye-bots gives the system more flexibility to adapt and find more efficient paths.

5.3 Experiments in more complex environments

We investigate the behavior of our system in more complex cluttered environments. A first one, shown in Fig. 5(a), has three large obstacles. Due to their size and their orientation perpendicular to the movement direction of foot-bots, they are difficult to pass using the foot-bots' imprecise obstacle circumnavigation system. Results for average travel time are shown in Fig. 5(b). The performance of static eye-bots is a lot worse than in the tests of Sect. 5.2, confirming that these obstacles are more difficult to pass. Nevertheless, the dynamic approach manages to get a performance that is close to that of a pre-defined efficient path.

Next, we place obstacles randomly between source and target. These obstacles are blocks of $1 \times 1\ m^2$. We use 0 up to 10 blocks. Fig. 6(a) shows an example scenario with 10 blocks. Results for our adaptive approach and for static eye-bots in a straight path are shown in Fig. 6(b). As the number of obstacles grows, the dynamic approach becomes more advantageous. Standard deviations are not shown, because differences between random scenarios lead to large variability between different runs (paired t-tests showed that the dynamic approach is better than the static one in each data point, with p-values in the order of 10^{-6}).

Finally, we explore the limitations of our system. In the scenario of Fig. 7(a), two large blocks are placed against the walls and form concave obstacles for foot-bots. The eye-bots' initial directional instructions based on IrRB communication lead foot-bots into a corner, and it is difficult for the foot-bots' obstacle circumnavigation to escape from this, as it tends to go the wrong way and rarely manages to find the way around the obstacles. This is confirmed by numerical results in Fig. 7(b): the performance of our approach is variable and far from that of the efficient path. Results could be improved using more sophisticated obstacle circumnavigation, but this would not solve the basic issue, that the path found by our system is dependent on the initial directions obtained from infrared communication. One solution is to let eye-bots explore directional instructions, by sending foot-bots in all possible directions, and learn the best instructions. In related work, we have investigated this for static eye-bots [16]. In future work, we will integrate this with the current work where eye-bots are mobile.



(a) (b)

Fig. 7. Concave obstacles experiments: (a) initial setup, (b) number of foot-bots vs. travel time (error bars show one standard deviation)

6 Conclusions and future work

We have investigated self-organized path finding in a heterogeneous robotic swarm consisting of two types of mobile robots, the eye-bots and the foot-bots. Eye-bots' role is to provide local guidance for foot-bot navigation. We studied how eye-bots can adaptively change their positions in order to improve the navigability of the path they indicate. We have shown how local interaction and

mutual adaptation between the two sub-swarms allows the system as a whole to find efficient paths in cluttered environments. In a number of experiments, we have investigated the performance of our system and have shown it can find efficient paths in a wide range of different scenarios. We have also identified the system's limitations, which are due to the dependence on wireless communication to get initial navigation instructions. In future work, we will work on this issue, by extending the system with capabilities to explore the environment.

References

1. Stirling, T., Wischmann, S., Floreano, D.: Energy-efficient indoor search by swarms of simulated flying robots without global information. Swarm Intelligence (in press)
2. Swarmanoid: Final swarmanoid hardware. Deliverable D13 of IST-FET Project *Swarmanoid* funded by the European Commission under Framework FP6 (2009)
3. Momen, S., Amavasai, B., Siddique, N.: Mixed species flocking for heterogeneous robotic swarms. In: Proceedings of IEEE Eurocon: The International Conference on Computer as a Tool. (2007)
4. Momen, S., Sharkey, A.: An ant-like task allocation model for a swarm of heterogeneous robots. In: Proceedings of the 2nd Swarm Intelligence Algorithms and Applications Symposium (SIAAS). (2009)
5. Pinciroli, C., O'Grady, R., Christensen, A., Dorigo, M.: Self-organised recruitment in a heterogeneous swarm. In: Proceedings of the 14th International Conference on Advanced Robotics (ICAR). (2009)
6. Fujisawa, R., Dobata, S., Kubota, D., Imamura, H., Matsuno, F.: Dependency by concentration of pheromone trail for multiple robots. In: Proc. of the 6th Int. Conf. on Ant Colony Optimization and Swarm Intelligence (ANTS). (2008)
7. Sugawara, K., Kazama, T., Watanabe, T.: Foraging behavior of interacting robots with virtual pheromone. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). (October 2004) 3074–3079
8. Vaughan, R., Støy, K., Sukhatme, G., Mataric, M.: Whistling in the dark: Cooperative trail following in uncertain localization space. In: Proceedings of the Fourth International Conference on Autonomous Agents, ACM Press (2000)
9. Wodrich, M., Bilchev, G.: Cooperative distributed search: The ants' way. Control and Cybernetics **26** (1997)
10. Goss, S., Aron, S., Deneubourg, J.L., Pasteels, J.M.: Self-organized shortcuts in the Argentine ant. Naturwissenschaften **76** (1989) 579–581
11. Batalin, M., Sukhatme, G., Hattig, M.: Mobile robot navigation using a sensor network. In: Proc. of the IEEE Int. Conf. on Robotics and Automation. (2004)
12. O'Hara, K., Balch, T.: Pervasive sensor-less networks for cooperative multi-robot tasks. In: Proceedings of the Seventh International Symposium on Distributed Autonomous Robot Systems (DARS-04). (2004)
13. Vigorito, C.: Distributed path planning for mobile robots using a swarm of interacting reinforcement learners. In: International Conference on Autonomous Agents and Multiagent Systems (AAMAS). (2007)
14. Bertsekas, D., Gallager, R.: Data Networks. Prentice–Hall (1992)
15. Swarmanoid: Final swarmanoid simulator. Deliverable D12 of IST-FET Project *Swarmanoid* funded by the European Commission under Framework FP6 (2009)
16. Ducatelle, F., Di Caro, G., Gambardella, L.: Cooperative self-organization in a heterogeneous swarm robotic system. Technical Report IDSIA-01-10, IDSIA (2010)