New task allocation methods for robotic swarms

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

***Abstract*— We study a situation where a swarm of robots is deployed to solve multiple concurrent tasks in a confined arena. The tasks are announced by dedicated robots at different locations in the arena. Each task requires a certain number of robots to attend to it simultaneously. We address the problem of task allocation: how can the robots of the swarm assign themselves to one of the announced tasks in a distributed and efficient way? We propose two novel methods: one relies on simple reactive mechanisms that are based on interaction through light signals, while the other uses a more advanced gossip-based communication scheme to announce task requirements among the robots. We evaluate both methods, and compare their performance. We also address scalability and robustness issues, in order to understand the usefulness of the methods in different swarm deployment conditions.**

I. INTRODUCTION

Swarm robotics is a form of collective robotics that takes its inspiration from social insects, such as colonies of ants, and from the related notion of swarm intelligence [20]. The central concept is to use large numbers of identical robots that individually have rather limited capabilities but when combined as a group are able to generate more complex behavior [18]. Swarm robotic systems work in a decentralized way and use only local control and communication. Typical properties of such systems are scalability, since the system can be extended to very large numbers of robots, flexibility, since robots can be dynamically added and removed, and fault tolerance, since individual robots are usually unimportant for the working of the system and there is no central point of failure [15].

In this paper, we address a problem of task allocation for robotic swarms. We consider a situation where a swarm of robots is deployed in a confined arena. Tasks appear at different locations in the arena and each task needs to be served by a certain number of robots simultaneously. The robots need to decide which task each of them will go to. The question we address is how this can be done efficiently in a distributed way, using only local communication.

We develop two mechanisms to deal with this problem. The first takes a very simple reactive approach. It is based on communication through light signals, whereby robots are attracted to one color of light and repulsed from another, in combination with random movements. The second mechanism is based on the explicit communication of structured information. When a task is announced, the number of robots it needs is communicated. This information is then passed on between the robots using a gossip mechanism, so that also robots further away can learn about it and react to it. In an evaluation study, we compare both mechanisms in terms of their efficiency, and we also investigate issues of scalability and robustness to communication failures.

In what follows, we first give a more detailed description of the problem we are addressing, and then discuss the related work in this area. Next, we describe the two task allocation mechanisms we propose. After that, we evaluate and compare the two systems. Finally, we draw conclusions and describe future work.

II. PROBLEM DESCRIPTION

The task allocation problem described here is situated in the broader context of a search task performed by a heterogeneous swarm consisting of two types of robots (this can also be seen as two separate swarms that collaborate). The first type are flying robots. They are called *Eyebots*. The second type are robots that move over the ground. They are called *Footbots*. Both types of robots are developed in the context of the EU-funded *Swarmanoid* project on heterogeneous swarm robotics [1], [2]. Images of these robots are shown in Figure 1. Within the Swarmanoid project, also a third type of robots is developed, the Handbots, which are left out of the discussion here to clarify the setup. In future work, they will perform part of the work that is here assumed to be done by the Footbots.

In the search task presented here, the heterogeneous swarm is requested to retrieve a particular target object from a room. To complete the task, the two types of robots

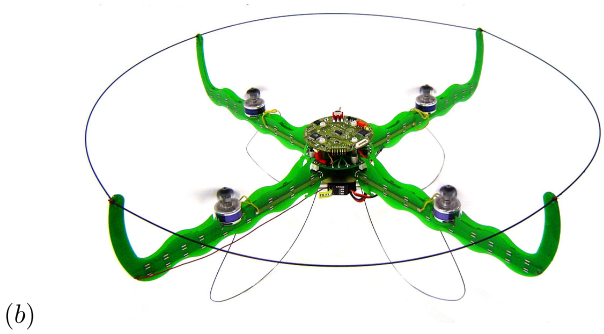

Fig. 1. Swarmanoid robots: (a) the Footbot (CAD draw) and (b) the Eyebot (prototype).

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 that might be made of data appearing in this publication.

The authors are with the Istituto Dalle Molle di Studi sull'Intelligenza Artificiale (IDSIA), Lugano, Switzerland. {frederick,alexander,gianni,luca}@idsia.ch.

cooperate. The flying Eyebots execute a high-level search, obtaining an overview of the room and identifying areas where the target object might be found (e.g., if the target object is a specific book, the Eyebots identify parts of the room where books are present). The Footbots then visit these areas of interest in order to execute a detailed search for the specific target object. This comes down to a two-level search procedure.

From the Footbots' point of view, the areas of interest indicated by the Eyebots are tasks that are announced at different locations in the environment. Here we are interested in the way the Eyebots announce the tasks and the way the Footbots react to this in order to get an efficient spreading of the Footbots over the tasks. The problem we address starts with a simultaneous announcement of multiple tasks by Eyebots at different locations in the arena, and finishes when all Eyebots have gathered enough Footbots around them to complete the local task.

III. RELATED WORK

A large number of strategies exist to solve task allocation problems in multi-robot systems. This is partly due to the high number of possible variations of the problem. An effort to formulate a taxonomy for existing task allocation problems was presented in [9]. According to this taxonomy, our work can be classified as a "single-task robots, multi-robot tasks, instantaneous assignment" problem.

One of the most popular approaches to deal with task allocation problems is the *market based strategy* (see [8] for an overview). In such a system, an auctioneer announces tasks, and robots make bids, indicating their cost or utility to deal with the tasks. Based on the different bids, the auctioneer then decides which robot will be assigned to which task. Market based task allocation combines the efficiency of a centralized approach (the auctioneer decides with overview of the situation) with advantages of distributed approaches (much of the calculation is done by the individual robots preparing their bids) [8].

For swarm robotics, market based systems are not always the most appropriate approach. This is because they use the auctioneer as a central decision maker, use an explicit assignment of individuals to jobs, and need some form of global or at least long distance communication. All of these elements reduce the scalability and robustness of the system, and conflict with the distributed and purely local way of working of the swarm paradigm (even though an auction-based system can still be the most efficient [11]).

Instead, the most commonly used task allocation strategies in swarm robotics are *threshold based systems*. These systems are based on observations of task or role allocation processes in social insects, whereby tasks, often implicitly, send out a signal, and the insects/robots react to this signal if it surpasses an internal threshold. Due to differences in this threshold among individuals, task allocation emerges in proportion to the task's signal intensity [12]. In [6], a variation of this mechanism is presented whereby all robots have the same threshold and the differentiation comes from the variability in the local observation of the signal intensity by the individuals. Several works address the issue of dynamically adapting the internal threshold, e.g. according to the estimated job density [5], according to the own past success rate for the task [13], or according to a combination of internal, external and social cues [14]. Other work combines this system with other methods for more complex task allocation [22].

Some of the work on task allocation in swarm robotics diverges from this threshold based mechanism. E.g., in [10], robots adapt their own task allocation based on the observed density of tasks and robots addressing them. In [16], the required target distribution of robots over tasks is known by all robots, and the authors investigate different strategies to get to this distribution, evaluating the amount of communication that is used in each one.

The work presented here is somehow naturally related to auction based systems, as the Eyebots announcing the tasks play a central role and could easily function as auctioneers. However, since we are interested in designing a task allocation method for a swarm robotic system, we want to avoid the bidding and explicit task assignment applied in auction approaches.

IV. TWO TASK ALLOCATION METHODS

In this section we describe the two task allocation methods we developed for swarm robotic systems. First we present the approach that uses light-based interaction, and next we describe the approach that is based on the exchange of structured information through gossiping. We assume that the Eyebots come down to ground level to announce the tasks to the Footbots.

A. Light-based task allocation

In the light-based task allocation system, Eyebots (and Footbots) use the multi-colored LEDs that are placed in a ring around their body [2] to influence Footbot behavior. The Footbots use their omnidirectional camera [2] to detect lights and react to them. The light-based task allocation system is built up around four basic behaviors:

- *Attraction to yellow light.* Footbots are attracted to yellow lights. When they see yellow lights in more than one direction, they go to the closest (they can estimate distance since it relates to vertical position in the field of view of the omnidirectional camera). Eyebots use yellow lights to attract Footbots to a task, as is shown in Figure 2(a). The number of yellow lights they use is proportional to the task size.
- *Repulsion from green light.* Footbots are repulsed from green lights: when they see green lights in the direction they are moving in, they turn away from it. In contrast to the attraction to yellow lights, repulsion is only active at a limited distance (set to about 50cm), so that the combination of a yellow and a green light attracts a Footbot up to a certain distance, after which the Footbot turns away. The repulsion behavior is used in two different ways. First, Eyebots show green lights in addition to the yellow lights, in order to control better the total number of Footbots they attract. This is shown in Figure 2(b). Second, Footbots that are attracted by yellow lights show green lights around, in order to repulse other Footbots from the tasks they are going to. This also limits the number of robots that come to serve a specific task. Moreover, it makes the Footbots that arrive at a task spread out. This is illustrated in Figure 2(c).

- *Internal frustration*. Each Footbot keeps an internal frustration level. This goes up whenever the Footbot experiences at the same time attraction and repulsion (as in the situation of Figure 2(c)), and goes down (but at a slower rate) when there is no repulsion. When the frustration reaches a fixed threshold, the Footbot executes an escape movement. This comes down to turning away from the direction in which attraction is observed and moving forward for a certain distance (enough to get outside the view of the attraction). This makes Footbots move away from events (points of attraction) that are being served by other robots (points of repulsion) and try other parts of the arena to find other tasks. The frustration mechanism is related to the internal motivations used in the Alliance architecture [17].
- *Random movements*. When none of the other three behaviors is active, the Footbots make random movements. These consist of turning in place for a random amount of time, and then moving forward for a random amount of time. This makes the Footbots execute a random search of the arena to find tasks.

The combination of these basic behaviors leads to a spreading of the Footbots over the different tasks, proportionally to the size of the tasks announced. An example is given in Figure 3, where the areas to be searched are bookshelves of different sizes. The three Eyebots indicate 1, 4 and 8 attracting lights respectively, attracting different numbers of Footbots. Note that the final number of Footbots is not necessarily exactly the same as the number of attracting lights: this number depends on the strength of the repulsion between the Footbots.

B. *Gossip-based task allocation*

The gossip-based task allocation system makes use of the *infrared range and bearing (Ir-RB) module* which is present on each Footbot and Eyebot [4]. This is an adaptation of the system presented in [19]. It consist of 26 infrared emitters and 16 receivers, placed all around the robot. Based on the quality of received signals, the system calculates an estimate of the relative bearing and range to other robots using the same system. The maximum range is 3 meters, and the precision is 20% for range estimates and 30 degrees for bearing estimates. The system also allows line-of-sight communication over the infrared signal with a nominal bandwidth of 40 Kbps. The advantage of this system is that received data can be related to information about the relative position of their sender.

The Eyebots use the Ir-RB system to send task announcement messages, in which they indicate the number of robots needed to complete the task. If they perceive Footbots nearby, they reduce this number. These task announcement messages are then forwarded by the Footbots and the other Eyebots in a gossiped way, i.e. each time they meet new neighbors, so that information about all tasks spreads among the swarm.

Each gossiped message contains information about all tasks a robot knows about. In detail, the following information is transmitted:

- *Robot ID*. The ID of the transmitting robot.

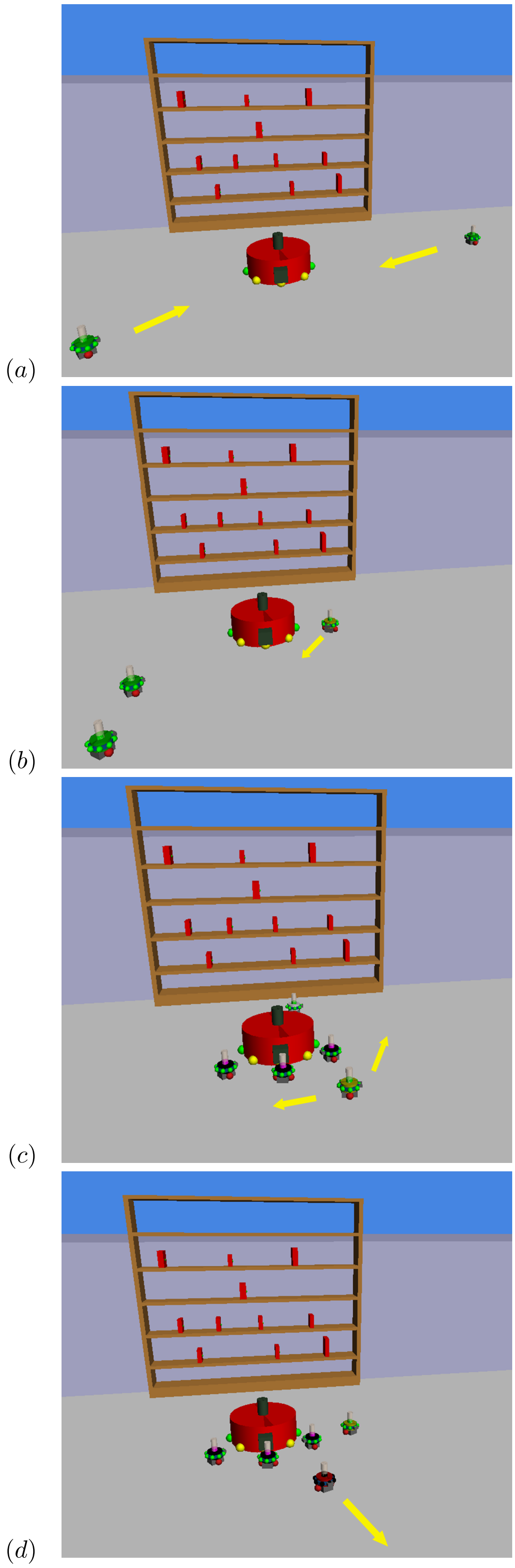

Fig. 2. Overview of the behaviors in the light-based task allocation mechanism: (a) attraction to yellow light, (b) repulsion from green light to get more precise placement, (c) repulsion from green light to fend off other robots, (d) evasive behavior when the frustration threshold is reached. In these figures, the color of the Footbot body illustrates its internal state: dark green means that it feels attraction, light green means repulsion, red means frustration, and black means that the Footbot is in place to perform the task. Yellow arrows show the movement direction of selected robots.

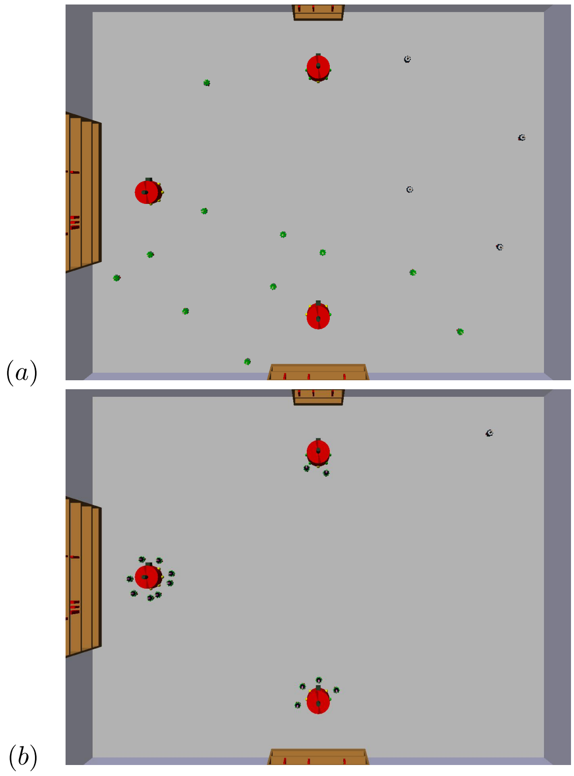

Fig. 3. An example of a task allocation scenario, (a) at the moment tasks are announced and (b) at the moment sufficient Footbots are assigned.

- *Number of tasks*. The number of tasks which the sender has received information about.
- *For each task*:
  - *Task ID*. This corresponds to the ID of the Eyebot announcing the task.
  - *Required workers*. The number of robots the task requires.
  - *Hops*. The number of hops (in terms of communication) to the task.
  - *Route length*. The distance to the task following the hops.
  - *Age*. The age of the information about the task.

When a robot receives information about a task, it needs to recalculate most of it before it can forward it in a message of its own. The number of required workers is decreased if the robot itself decides to go towards this task. The number of hops is increased by 1. The route length is increased by the distance to the robot the message was received from. Finally, the age value is increased. If the task age exceeds a threshold the task information is discarded and not re-sent in the next time-step.

The information about the tasks is used by the Footbots to decide on their actions. In general, the Footbot has four different behaviors:

- *Attraction to next task hop*. The nearest task is defined using the number of hops as first criterion and the route length as second criterion. The attraction of the task is only active when the number of additionally required robots for this task is greater than 0. The robot goes towards the next hop of the task (i.e., the robot it received the task information from), using the bearing information from the Ir-RB system. It steers on a circular path around the robot when it is close to it, until it sees the following hop. The reason to go hop by hop rather than straight to the task is to find obstacle free paths (since the Ir-RB communication only works over line-of-sight).
- *Internal frustration*. The robot has an internal frustration level value for each known task. This level increases with its distance to the task and with the number of robots that are near the task. The frustration decreases each time step with a small amount. When the frustration for a task passes a certain threshold, the robot will not go to this task.
- *Random movements*. This behavior is active when the robot does not know any task to go to (i.e., it knows only tasks that have enough robots or which it has a high frustration level for). The robot steers to a random position in its surrounding area. When the robot reaches this position or detects an obstacle on its way, another random position is generated.
- *Obstacle avoidance*. The obstacle information is based on proximity sensor values. When an obstacle is detected, a motion force in the opposite direction of the obstacle is added to the intended movement of the robot. When the robot is very close to the task itself the obstacle avoidance behavior is suppressed to stabilize already aggregated robots to tasks.

Compared to the light-based system described in Subsection IV-A, the gossip-based task allocation system is a bit more complex, as it requires the exchange and processing of structured information. However, unlike a market-based system (see section III) the task allocation is still entirely based on autonomous decisions of the individual robots, and no explicit assignment of robots to tasks is needed. An advantage compared to the light-based system is that information about tasks is disseminated over larger distances. Moreover, the fact that we use gossiping entails that the information can spread over the network of robots using only local message exchanges, without the need for full connectivity at any time. This way, information is flowing opportunistically between robots, which is an important advantage in sparse networks [23].

V. EVALUATIONS AND COMPARISONS

In the following we evaluate and compare the two task allocation methods. All tests are done using the Swarmanoid simulator [3], which uses the Open Dynamics Engine library [21] for the calculation of the physical movements and collisions of the robots and their environment, and the OpenGL library [7] for visualization.

We carried out experiments using three different setups, as shown in Figure 4: (a) an open environment (room size $6 \times 10m^2$), (b) an environment with obstacles (room size $6{\times}10m^2$), and (c) a maze (room size $9{\times}9m^2$). In setups (a) and (b), there are three Eyebots announcing tasks, whereby one task requires one Footbot, the second three, and the last five. In setup (c) there are two Eyebots announcing tasks, one of which requires four Footbots, and the other five. We carry out tests with increasing numbers of Footbots in the room: from 10 up to 40.

In the first place we are interested in efficiency: how quickly can the different task allocation systems assign the correct number of robots to each task. Figure 5

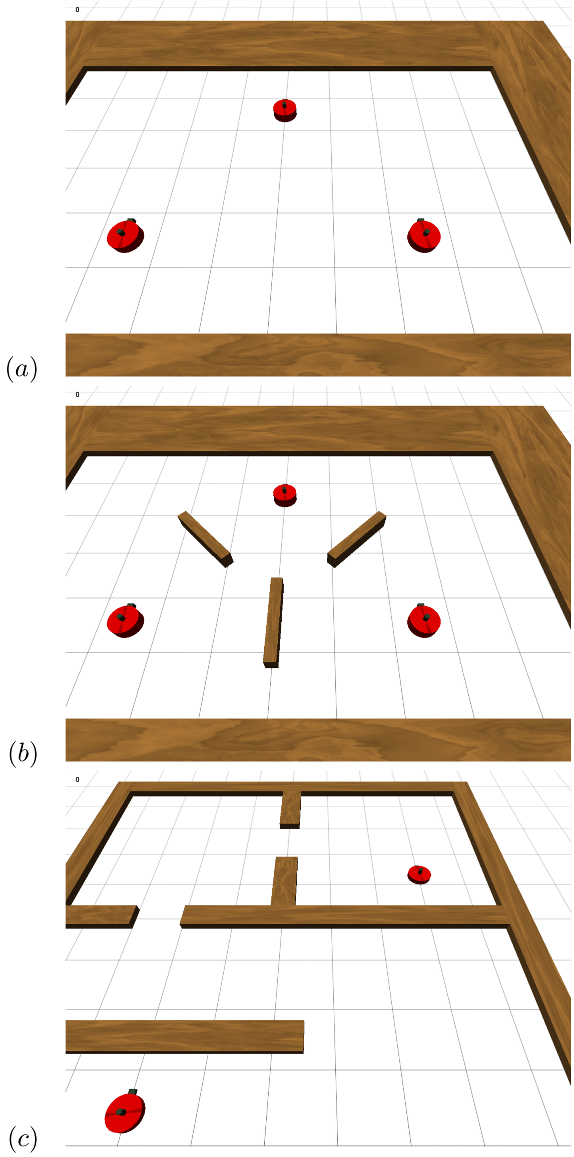

Fig. 4. The three different test setups we used: (a) an open environment, (b) an environment with obstacles and (c) a maze.

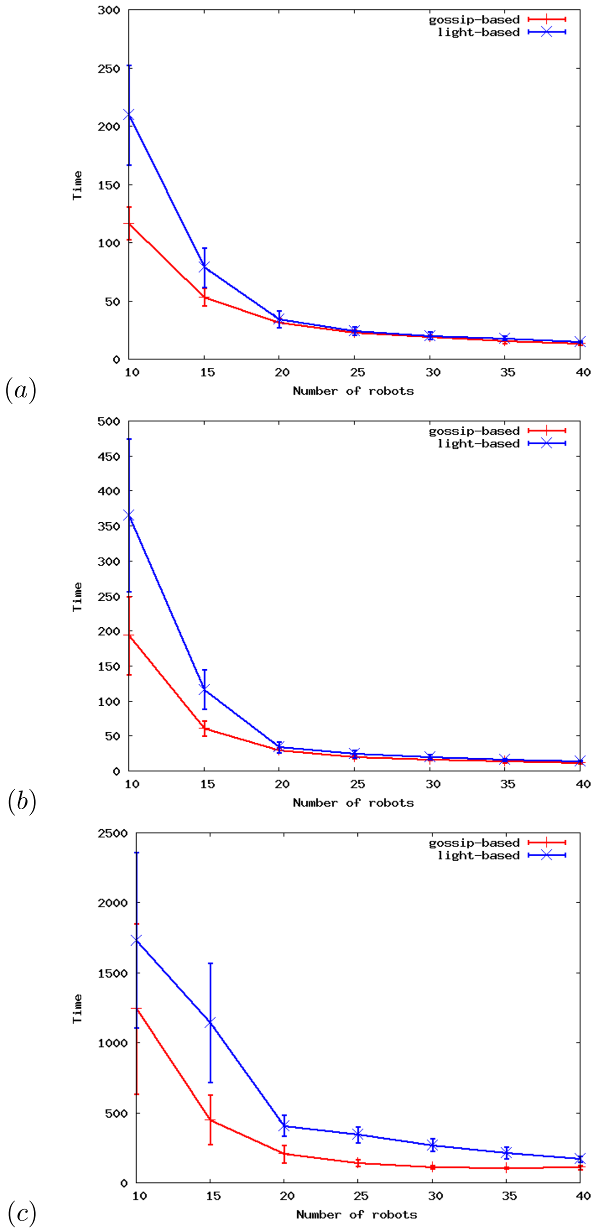

Fig. 5. The time needed to obtain the required allocation of robots to tasks in the three different setups, with increasing numbers of robots. Error bars show a 90% confidence interval using a t-test.

shows the time needed to reach the desired configuration in each of the three setups, with error bars indicating a 90% confidence interval. In all three setups, and for both algorithms, the performance improves with increasing numbers of Footbots, as more robots are available to serve the different tasks. A number of 20 robots or more seems to guarantee good performance in all setups, while especially in the maze setup performance suffers considerably for lower numbers of robots. In general we can see that the gossip-based task allocation method gives better results than the light-based approach: the presence of structured information propagated between the robots allows to make better decisions. The difference between the two methods decreases with an increasing number of Footbots though, and becomes insignificant for 20 robots or higher both in the open and the cluttered environment.

Despite its good performance, the gossip-based task allocation method has an important drawback compared to the light-based method: it requires the use of wireless communication in order to exchange information. Different from light signals, wireless packet transmission over the Ir-RB system may fail due to interference with other transmissions or with other infrared signals in general (e.g., signals produced by distance and proximity sensors). Interference increases with growing numbers of robots, and hence limits the scalability of the system.

Precise evaluations of the level of interference are not possible through our simulation system, and will therefore be addressed in later tests on the real robots. Here, we focus on the effect that the loss of communication packets has on the performance of the system, i.e. how robust the gossip-based system is with respect to packet loss. Figure 6(a) shows the bandwidth consumption per robot for the setup of Figure 4(b) with 15 Footbots, when only a fraction of the scheduled messages are sent (fraction 1 means 1 packet every control step of 100ms, while fraction 0.01 means 1 packet per 10 seconds). While these levels of bandwidth can nominally be supported by the Ir-RB system, it is expected that competition for the wireless channel and the

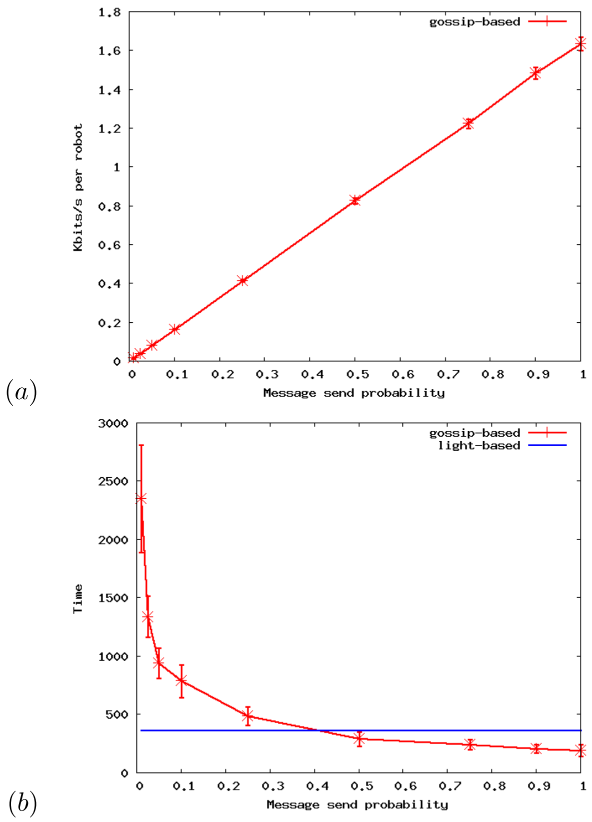

Fig. 6. For the setup of Figure 4(b) with 15 Footbots, (a) the bandwidth consumption and (b) the time required to obtain correct task allocation, when only a fraction of the messages are sent (e.g., 0.5 means that 50% of the scheduled messages are sent). The performance of the light-based approach in the same setup is also shown.

lack of centrally controlled medium access control will lead to packet loss even for a relatively low number of robots. Figure 4(b) shows the performance of the gossip-based system when only a fraction of the packets are sent. As can be seen, the performance suffers considerably when less than 25% of the messages are sent. At that point, the light-based approach becomes preferable.

In summary, the gossip-based approach works better than the light-based approach when the density of robots is low and operating conditions are good. However, it scores less good for scalability and robustness to communication failures, which are important issues in swarm robotics. Further tests on the real system have to indicate how severe these problems can be.

VI. CONCLUSIONS AND FUTURE WORK

In this paper we have described two task allocation mechanisms for swarm robotic systems. The first is based on light signalling between robots, while the other relies on the gossiped exchange of structured information about tasks. Neither of the algorithms applies an explicit allocation of specific robots to tasks. In comparison tests, we found that the gossip-based algorithm is more efficient than the light-based algorithm in highly cluttered environments or when the total number of robots is low, while the difference between the algorithms is insignificant when more robots are deployed and the environment is not too complex. We also found that the gossip-based approach has limited robustness to packet loss and may therefore be less scalable. As a consequence, the light-based approach might be more appealing in large swarm robotic systems.

In future work we will complement the results we got via simulation with tests using the real Swarmanoid robots, in which we will explicitly investigate issues of interference, robustness to packet loss and scalability. Also, we want to investigate how to integrate the two different task allocation methods, so that the swarm can switch between them according to the deployment scenario, and get the best of both systems. Finally, we want to extend these mechanisms to the full Swarmanoid system, which means including the third kind of robot, the Handbot.

REFERENCES

[1] Swarmanoid: Towards humanoid robotic swarms. `http://www.swarmanoid.org`.

[2] Hardware design. Internal Deliverable D4 of IST-FET Project *Swarmanoid* funded by the European Commission under the FP6 Framework, 2007.

[3] Simulator prototype. Internal Deliverable D7 of IST-FET Project *Swarmanoid* funded by the European Commission under the FP6 Framework, 2008.

[4] Swarmanoid prototype evaluation and redesign. Internal Deliverable D11 of IST-FET Project *Swarmanoid* funded by the European Commission under the FP6 Framework, 2008.

[5] W. Agassounon and A. Martinoli. Efficiency and robustness of threshold-based distributed allocation algorithms in multi-agent systems. In *Proceedings of the First International Joint Conference on Autonomous Agents and Multi-Agent Systems AAMAS-02*, 2002.

[6] W. Agassounon, A. Martinoli, and R.M. Goodman. A Scalable, Distributed Algorithm for Allocating Workers in Embedded Systems. In *Proc. of the IEEE Conf. on System, Man and Cybernetics*, 2001.

[7] OpenGL Architecture Review Board. *OpenGL(R) Reference Manual: The Official Reference Document to OpenGL, Version 1.4*. Addison-Wesley Professional, 2004.

[8] M.B. Dias, R. Zlot, N. Kalra, and A. Stentz. Market-based multirobot coordination: A survey and analysis. *Proceedings of the IEEE*, 94(7), July 2006.

[9] B.P. Gerkey and M.J. Matarić. A formal analysis and taxonomy of task allocation in multi-robot systems. *International Journal of robotics Research*, 23(9), September 2004.

[10] C. Jones and M.J. Mataric. Adaptive division of labor in large-scale minimalist multi-robot systems. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, October 2003.

[11] N. Kalra and A. Martinoli. A comparative study of market-based and threshold-based task allocation. In *Proceedings of Distributed Autonomous Robotic Systems (DARS)*, July 2006.

[12] M.J.B. Krieger and J.-B. Billeter. 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.

[13] T.H. Labella, M. Dorigo, and J.-L. Deneubourg. Self-organized task allocation in a group of robots. In *Proc. of the 7th Int. Symp. on Distributed Autonomous Robotic Systems (DARS)*, 2004.

[14] W. Liu, A.F.T. Winfield, J. Sa, J. Chen, and L. Dou. Towards Energy Optimization: Emergent Task Allocation in a Swarm of Foraging Robots. *Adaptive Behavior*, 15(3), 2007.

[15] A. Martinoli. Collective complexity out of individual simplicity. *Artificial Life*, 7(3), 2001.

[16] J. McLurkin and D. Yamins. Dynamic task assignment in robot swarms. In *Proceedings of Robotics: Science and Systems*, 2005.

[17] L.E. Parker. Alliance: An architecture for fault tolerant multi-robot cooperation. *IEEE Trans. on Robotics and Automation*, 14(2), 1998.

[18] L.E. Parker. Distributed intelligence: Overview of the field and its application in multi-robot systems. *Journal of Physical Agents*, 2(1), 2008.

[19] J. Pugh and A. Martinoli. Relative localization and communication module for small-scale multi-robot systems. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2006.

[20] A.J.C. Sharkey. Swarm robotics and minimalism. *Connection Science*, 19(3), September 2007.

[21] Russell Smith. *Open Dynamics Engine v0.5 User Guide*, 2006.

[22] D. Zhang, G. Xie, J. Yu, and L. Wang. Adaptive task assignment for multiple mobile robots via swarm intelligence approach. *Robotics and Autonomous Systems*, 55(7), 2007.

[23] Zhensheng Zhang. Routing in intermittently connected mobile ad hoc networks and delay tolerant networks: overview and challenges. *Communications Surveys and Tutorials, IEEE*, 8(1):24–37, 2006.