J Intell Robot Syst (2016) 82:325–337
DOI 10.1007/s10846-015-0277-0

Collaboration in Multi-Robot Exploration: To Meet or not to Meet?

Torsten Andre · Christian Bettstetter

Received: 31 March 2015 / Accepted: 15 September 2015 / Published online: 26 September 2015

Abstract Work on coordinated multi-robot exploration often assumes that all areas to be explored are freely accessible. This common assumption does not always hold, especially not in search and rescue missions after a disaster. Doors may be closed or paths blocked detaining robots from continuing their exploration beyond these points and possibly requiring multiple robots to clear them. This paper addresses the issue how to coordinate a multi-robot system to clear blocked paths. We define *local collaborations* that require robots to collaboratively perform a physical action at a common position. A collaborating robot needs to interrupt its current exploration and move to a different location to collaboratively clear a blocked path. We raise the question *when* to collaborate and *whom* to collaborate with. We propose four strategies as to when to collaborate. Two obvious strategies are to collaborate immediately or to postpone any collaborations until only blocked paths are left. The other two strategies make use of heuristics based on building patterns. While no single strategy behaves optimal in all scenarios, we show that the heuristics decrease the time required to explore unknown environments considering blocked paths.

Keywords Collaboration · Robot exploration · Mobile robot teams · Indoor exploration · Multi-robot systems · Autonomous systems

1 Introduction and Motivation

Coordination in multi-robot systems during the exploration of unknown environments has been widely discussed in the literature [12, 17]. Though exploration missions in which robots have to interact with the environment to continue exploration, especially ones where multiple robots have to collaborate, have received less attention. Heterogeneous robots may collaborate to make use of their individual specializations [9] while homogeneous robots may join forces to overcome a single robot's limitations [18]. In search and rescue missions, examples include clearing a path by opening doors, moving debris, extinguishing fires, giving first aid to casualties or retrieving them. We denote such collaborative efforts as *local collaborations*. The term "local" indicates that robots need to be in their immediate physical neighborhood, i.e., a helping robot has to interrupt its current task and travel to a requesting robot. Upon completion of the collaboration, the helping robot may resume its prior task. Local collaborations have an impact on exploration time, travel distance, and wireless connectivity.

T. Andre (✉) · C. Bettstetter
Networked and Embedded Systems, Alpen-Adria-Universität Klagenfurt, Lakeside Park B02a,
Klagenfurt, Austria
e-mail: torsten.andre@aau.at

C. Bettstetter
Lakeside Labs GmbH, Lakeside Park B04b,
Klagenfurt, Austria

This paper discusses two factors influencing the decision on collaboration, namely *when* to collaborate and *whom* to collaborate with. For example, consider the floor plan depicted in Fig. 1. Given a partially explored environment, robot r_3 reaches a location requiring collaboration to get to the point P_1 because the direct path is blocked. It has to decide whether to request collaboration and if so when to do it. A well-founded decision cannot be made with the currently available information. If r_2 continues exploration, r_2 will reveal an alternative path to P_1, thus making collaboration unnecessary. If no other path exists, however, postponing the collaboration needlessly prolongs the exploration time. Here, r_3 is assumed to start a collaborative action by broadcasting a collaboration request. The robot requesting collaboration is called r_C, i.e., $r_3 \equiv r_C$. The robots receiving the request have to determine their suitability for the collaboration: r_1 has to decide whether to abort the exploration of a room with no more paths to follow; r_4 and r_5 partition the communication network because they are the only robots connecting r_6; r_6 has the longest distance to travel.

The impact of the influencing factors *when* and *whom* can be decomposed into two problems. First, robots need to decide whether to request collaboration or to continue exploration. Second, if a collaboration is requested a helping robot needs to be determined. The selection of a helping robot can be formulated as an assignment problem in the second step.

The first problem, deciding when to collaborate, becomes part of the autonomous multi-robot exploration problem. That is the problem of multiple robots exploring an unknown environment while typically creating a map thereof. Juliá *et al.* identify autonomous exploration as an instance of the partially observable Markov Decision Process (POMDP) with continuous states [17]. A Markov Decision Process (MDP) is defined as a 5-tuple $\left(S, A, p(s'|s,a), r_a(s), \gamma\right)$ where S is a set of system states, A a set of possible actions, $p(s'|s,a)$ a transition model for a state to change from $s \in S$ given action $a \in A$ to $s' \in S$, a reward $r_a(s) \in \mathbb{R}$ obtained when executing action $a \in A$ when in state $s \in S$, and $\gamma \in [0, 1)$ a discount factor for delayed rewards. The goal is to find a policy π which determines the action $\pi(s) = a$ when in state s maximizing the gained rewards over a horizon n:

$$\sum_{t=0}^{n} \gamma^t R_{a_t}(s_t, s_{t+1})\,. \tag{1}$$

Using the horizon of the MDP, a decision whether to collaborate now or later may be obtained given the expected rewards. However, in robot explorations (especially for local collaborations), the rewards and transitions are initially unknown to the robot and must be learned. Reinforcement learning allows to determine rewards and transitions but has limitations. The reward of a collaboration significantly depends on the environment and the current state of its exploration. Little progress on the exploration may have significant impact on the reward of a collaboration. Consider the scene depicted in Fig. 1. The reward for a collaboration initiated by r_3 seems high. But shortly after r_2 has moved around the corner, the reward of a collaboration vanishes because an alternative path is unveiled.

Semantic information can help to determine a possible reward. Consider the entrance to an office to be blocked. Assuming the majority of offices has only a single entrance, collaboration is required to access the office and should be performed immediately. In comparison, exhibition rooms in a museum would probably be expected to have at least a sec-

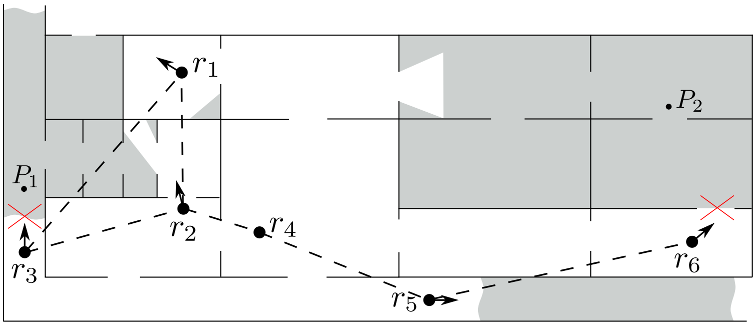

Fig. 1 *White* area represents explored space, *gray* unexplored. *Red crosses* indicate collaboration points. The robots form a wireless ad hoc network. The *dashed links* indicate connectivity between robots. Robots share the map. r_3 becomes the requesting robot r_C once it starts a collaboration

ond entrance, i.e., time consuming collaboration may be postponed due to an alternative access path. But current semantic place recognition does not yet allow reliable detection [10].

Closed form solutions to POMDP are computationally expensive and do not scale well [17]. Therefore, the majority of literature on multi-robot exploration considers the problem as an assignment problem. Frontiers, i.e., transitions between known and unknown space, are assigned to robots based on the information available at time of assignment. Robots then travel to these frontiers to increase the knowledge of the environment. Assignment is performed with the intention to minimize exploration time. Constraints such as travel distance [5], path safety [11], connectivity [19], and map quality [14] may be considered during assignment.

We focus on information which can be obtained reliably for current robot systems and do not consider semantic knowledge. Due to the complexity of the problem, we derive heuristics which we test in random environments not to bias the analysis. The paper contributes to the field of collaborative multi-robot exploration by

- defining the problem of local collaboration,
- discussing influencing factors, and
- analyzing the performance of four collaboration strategies by simulation.

2 Related Work

Many papers discuss collaboration in multi-robot systems during explorations [6, 7, 17, 24]. The majority of papers, however, focuses on coordinating the assignment of frontiers to robots in order to decrease exploration time. Only few papers focus on how collaborations between robots affect the exploration mission itself. Singh and Fujimura consider a collaborative system where larger robots request the assistance of smaller robots for places they cannot reach [22]. Wurm *et al.* consider marsupial robot teams [23], where carrier robots deploy smaller robots to explore the environment. Other forms of collaboration include managing network connectivity. Pei *et al.* consider robots to relay information between a base and exploring robots [21], and de Hoog *et al.* consider data mules in case direct connectivity cannot be reached [16]. Nevatia *et al.* make use of experienced personnel to coordinate their partially autonomous system outperforming a complete autonomous system. They consider a search and rescue mission including humans in the organization of a multi-robot system [20]. But in case of major disasters, e.g., earthquakes, floods, hurricanes, fully autonomous robot systems could free skilled personnel required for different tasks.

In all cases it is either assumed that paths are available to complete the exploration or that the mission's sole objective is the exploration. In contrast, possible use cases as motivated earlier have not been addressed so far.

3 Local Collaborations

We start by formally defining the problem of local collaborations described informally above.

Definition 1 For a set R of robots, *local collaboration* is the process of a robot $r_i \in R$ to request one or more robots $r_j \in R \setminus \{r_i\}$, $j = (1, \ldots, |R| - 1)$, to join in the location of r_i to collaboratively perform an action.

A local collaboration may be classified by its relative importance as to when collaboration is required:

1. *Critical and urgent*: Collaboration is required immediately (without prioritization of objectives),
2. *Critical and non-urgent*: Collaboration is mission critical and must be performed before the mission goal can be completed, but can be postponed to a later point in time (with prioritization of objectives),
3. *Optional*: Collaboration is optional and only required under certain circumstances. For example, clearing a path to a room may only become mandatory if no other path can be found.

To give an example, these three classes reflect procedures applied by firefighters [8]. In case of a burning building, for example, as soon as casualties have been found they are rescued due to the inherent risk of the fire to spread (class 1). Rescuing of casualties may only be postponed if one knows about people being trapped with higher risk (class 2). If no initial information is obtained concerning potential positions of casualties, firefighters have to decide whether to clear a path at all (class 3).

This paper considers the third class. The decision if and when to collaborate is the most challenging having the most degrees of freedom. It does not necessarily require collaborations and allows to postpone them until they may become obsolete. Possible additional classes depending on a given application (such as optional, but immediate collaboration) can be constructed combining aspects of the aforementioned classes. For illustration, we consider the case in which collaborations are required to clear a path to continue exploration. Paths may be cleared by jointly moving debris, opening doors, or other means.

3.1 When to Collaborate?

Definition 2 An *environment* is modeled as an undirected graph $G = (V, E)$ with a set of vertices V connected by a set of edges E. A set of robots R explores an unknown environment G creating a map $G' = (V', E')$. A vertex $v \in V$ is added to V' once v is visited by a robot. An environment is said to be *completely explored* if all vertices $v \in V$ have been visited, i.e., $V = V'$.

Definition 3 The subset $L \subseteq E$ denotes *obstructed edges* requiring local collaborations of two or more robots to make them passable. Once a local collaboration is performed on edge $l \in L$, l is removed from L and thus becomes passable.

In the following we consider two collaborating robots.

Definition 4 A *strategy* S defines when to collaborate and whom to collaborate with. We will define and compare four different strategies.

Problem 1 Let T be the time to complete exploration of an unknown environment, let $t = [0, \ldots, T]$ be the mission time, $l_i \in L, i = 1, \ldots, |L|$ the obstructed edges with $l_i \neq l_j$ for $i \neq j$, and t_{l_i} be the time when robots collaborate to clear $l \in L$. The problem is to determine when to collaborate to minimize T, i.e.,

$$\underset{t_{l_1},\ldots,t_{l_{|L|}}}{\arg\min}\ T\left(G, R, S(t_{l_1}, \ldots, t_{l_{|L|}})\right). \qquad (2)$$

In case of optional collaborations, it is not just a question when to collaborate but whether to collaborate at all. If a collaboration for an obstructed edge $l \in L$ turns out to be obsolete, we set $t_l = \infty$.

The decision on whether to collaborate or not becomes a question as to when sufficient information is available to deduce required collaborations. For example, consider r_6 in the map excerpt depicted in Fig. 1. Robot r_6 has to decide whether to request collaboration to clear the path or whether another yet undiscovered path may be found. An obvious choice to deem a collaboration necessary is to postpone any collaborations until only blocked paths remain. If areas behind a blocked path have not yet been explored, collaboration is required. But in comparison to r_3, no other robot may unveil an alternative path in the foreseeable future. If no alternative path to P_2 was found, comparably high travel costs would be required to return to r_6's current position to clear the path. If an alternative path to P_2 was discovered, an unnecessary collaboration would waste time and resources.

In order not to have to explore the whole area around a point of interest, we consider different types of patterns found in buildings [1] and propose two heuristics applicable by robots to determine when to collaborate. The first heuristic makes use of repetitive building structures. For example, in a multistory building, robots may use already explored floors as a reference map when exploring other floors assuming similar or identical structure between the two floors. Other examples include identical apartments in apartment or residential houses. Having picked a frontier with a blocked path, robots decide to collaborate if no other path can be found between a robot's current position and the frontier in the reference map. We assume a topological world representation where vertices V represent rooms and edges E paths between rooms.

Heuristic 1 *An environment G consists of disjunct parts $G_i \in \{G_1, G_2, \ldots, G_N\}$ with corresponding V_i, E_i with identical or similar structure. Robots having completed G_j progress to the unknown G_k ($j, k = 1, \ldots, N; j \neq k$). G_j and G_k have identical structure and robots are assumed to be aware of this. Let $v_k(1), v_k(2) \in V_k$ be two rooms connected by an obstructed edge $l \in L$ of which one is a frontier and let $v_j(1), v_j(2) \in V_j$ be the corresponding pair of rooms* *in G_j. Further, let the set $P_{v_j(1),v_j(2)}$ be all known routes between $v_j(1)$ and $v_j(2)$. A robot will immediately request collaboration if $|P_{v_j(1),v_j(2)}| = 1$, i.e., if only one path is known to the frontier. Otherwise, a robot will postpone the collaboration expecting to find an alternative route which may not be obstructed. In case only obstructed paths are available, the strategy selects the path with the least number of alternative paths, i.e.,* $\arg\min_{v(1),v(2)} |P_{v(1),v(2)}|$.

Note that robots do not have partial knowledge of the environment but belief thereof. Two floors may be completely different so that yet unexplored space is assumed to be unknown. The drawbacks of this heuristic are as follows: firstly, it requires parts of an environment to be already explored and, secondly, it is not universally applicable requiring building patterns on a large scale. The second heuristic makes use of smaller patterns in the form of repetitive room structures, e.g., an office building where rooms often have a similar structure with a single door facing to the corridor. The degree of a vertex $v \in V'$ models the number of paths between a room v and other rooms. From the distribution of degrees robots may infer the probability to find an alternative path to a blocked one. The distribution is estimated considering already explored rooms.

Heuristic 2 *Let X be a random variable modeling the degree of vertices $v \in V'$, x_v the number of already known blocked paths into a room $v \in V'$, and $p \in [0, 1]$ a threshold. A robot will request collaboration if either $P[x_v \geq X] \leq p$ or no unblocked paths are left.*

3.2 Whom to Collaborate with?

Whenever a robot requests support for collaboration, a helping robot needs to be determined.

Problem 2 Let r_C be a robot requesting collaboration. A helping robot r is determined by maximizing a utility value $u(r)$, i.e.,

$$\arg\max_{r \in R \setminus \{r_C\}} u(r) \tag{3}$$

In the following we discuss the impact of travel overhead, preemption of exploration, and connectivity between robots.

3.2.1 Travel Overhead

A collaborating robot has to travel to a requesting robot for collaboration. The travel distance between these robots is considered to determine the suitability of the helping robot [17]. The shorter the travel distance the better.

3.2.2 Preemption of Current Exploration

We discuss whether robots should abort their current exploration task immediately once collaboration has been requested. Consider r_3 in Fig. 1 requesting a collaboration. Robot r_1 is close to complete the exploration of a room which does not lead to any yet unexplored rooms. If r_1 completes the exploration before departing for the collaboration, it will not have to return to same room. Otherwise, r_1 will have to come back if it departs immediately for collaboration spending additional resources while traveling.

González-Baños and Latombe suggest to derive the information gain I corresponding to a frontier's size to determine their relative importance [14]. Instead of the gain, we consider the approximated time $\widetilde{T}(r) \in (0, \infty)$ it takes robot r to explore its current frontier. The time is approximated by the size of the frontier and is finite if the frontier is completely surrounded by obstacles or already explored space. For example, the yet unexplored space close to r_1 in Fig. 1 is bounded. Therefore, $\widetilde{T}(r_1)$ is finite. In comparison, the size and thus the exploration time for frontiers of r_2, r_3, r_5, and r_6 cannot be determined. We set $\widetilde{T}(r) = \infty$ for $r \in r_2, r_3, r_5, r_6$.

Accordingly, let $\widehat{T}(r) \in [0, T)$ be the approximated travel time for robot r to reach the requesting robot r_C. The time is approximated by the travel distance and the robot speed. We define $u_p(r) = \widetilde{T}(r)/\widehat{T}(r)$ as the preemption value. Robots are ranked to participate in collaboration in descending order of $u_p(r)$. Robots with infinite $\widetilde{T}(r)$ or $\widehat{T}(r) = 0$ are selected first ($u_p = \infty$). Additionally, by setting a threshold for u_p the overhead of travel time to exploration time can be controlled. Robots preempt their exploration in favor of collaboration only if u_p is above a threshold. For example, a threshold of $\frac{1}{2}$ stalls collaborations if the travel time is twice the exploration time. Therefore, the helping robot has time to complete exploration of its current frontier. After the collaboration, it may continue at a new frontier and does not have to return to its previous location.

3.2.3 Connectivity

Wireless connectivity is important for at least two reasons. Firstly, robots must be able to request collaborating partners. It may be of great importance to reach a robot with capabilities required for a specific task, especially in heterogeneous systems. Secondly, coordination can have a big impact on exploration time [2] requiring some form of communication. The selection of a collaborating robot may have a big impact on network connectivity. The robot selected for collaboration travels from its current location to the location of r_C, thus misses in the communication network. The intention is to identify the robot whose movement will decrease the connectivity of the network least. Consider the network depicted in Fig. 2a. The edges indicate connectivity between robots. r_4 is selected for collaboration, i.e., travels to r_C's location. Figure 2b illustrates the network once r_4 has reached r_C and how the network is disconnected.

We use the Fiedler value as a measure for network connectivity [13]. Maximizing the Fiedler value leads to increased network connectivity. A Fiedler value of zero indicates a disconnected network allowing to derive articulation points. We determine the impact of a robot on network connectivity by removing one robot at a time from the network and computing the resulting Fiedler value. The most suitable candidate is r_6 because its removal has no impact on the resulting network connectivity. r_1 is preferred to r_3 because the removal of r_3 reduces the number of paths between the network's left hand side and its right. r_4 and r_5 are identified as articulation points. Their removal disconnects the network and, therefore, are the least preferable robots to select. We only select robots which are not articulation points for local collaborations to not partition the network.

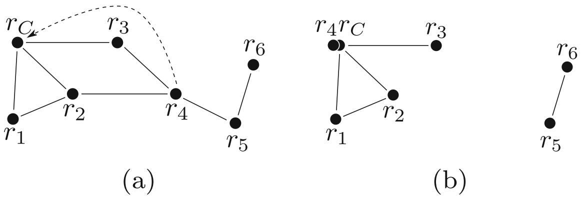

Fig. 2 Network connectivity between robots **a** before and **b** after r_4 has reached the requesting robot

4 Strategies and Simulation Model

4.1 Collaboration Strategies

We refine the previous definition of a *strategy* to explore an unknown environment to distinguish four different strategies: The strategies S_{H1} and S_{H2} implement the strategies defined in Heuristic 1 and 2, respectively. Further, we compare them to the *Immediate* strategy S_{I} which calls for immediate collaboration once an edge $l \in L$ is discovered and no alternative path is already known. The strategy *Postpone* S_{P} delays collaborations as long as possible; robots are not engaged in any collaborative actions as long as there are freely accessible, unknown areas.

The collaboration threshold p for S_{H2} is set to 0.15. Note that $p = 1$ equals S_{I}, $p = 0$ equals S_{P}. Collaborations are stalled until the collaborating robot has finished its exploration if $u_p(r) \leq 1$, i.e., if a robot spends the same time traveling back and forth to the collaboration as it requires to finish its exploration, it stalls the collaboration. For S_{H2} we assume the robot is aware of the exact room layout, though unaware of possibly required collaborations. This leads to an upper bound of the strategy's performance.

Additionally we compare the four strategies with the case W without the need for collaborations, i.e., all paths are freely accessible, to illustrate the impact of local collaborations on system performance.

4.2 Simulation Model

The focus of the evaluation is set to comparing the behavior and performance of the different strategies. We implement a dedicated simulator. Simulation of the strategies using the Robot Operating System (ROS) and publicly available ROS packages for coordinated multi-robot exploration [4] fail to perform this task. The mapping and map merging for large environments are not yet robust enough to obtain meaningful results with respect to the local collaboration strategies. Planning, navigation, and map errors prevent robots from completing the exploration process or to navigate to each other. Instead, we focus on comparing the strategies in the absence of above errors without loss of generality. All strategies depend on a reliable global map to be able to navigate towards each other. Therefore, mapping or positioning errors have the same effect on all strategies. Robots are assumed to be able to position themselves reasonably well and exchange maps. With respect to the heuristics robots are capable of detecting paths between rooms and are aware of repetitive structures G_i. For the simulation of S_{H1} we assume an identical floor plan, i.e., robots are assumed to already have explored a part G_0 when continuing to G_1. The analyzed exploration times only consider the time to explore G_1. While this is unfair with respect to the other strategies, it gives an upper bound on the performance of S_{H1}.

The environment and its layout have significant impact on the performance of a collaboration, i.e., which paths are available between rooms and which are blocked. Scenarios in which one strategy outperforms the other can easily be constructed. To avoid biasing the analysis of the strategies, the robots operate in randomly generated, unknown environments. Robots are capable of detecting rooms represented by vertices $v \in V$ positioned in a grid connected with a varying number of entrances per room. Entrances are blocked with a given blocking rate. Entrances and collaboration points are positioned randomly. Figure 3 illustrates two randomly generated environments G with a total of 16 rooms each. Tests with larger environments yield qualitatively comparable results.

We assume all collaborative actions to take 50 time steps corresponding roughly to the exploration of two rooms, which are of size 25 square units each. While action times may be expected to have different durations in real deployments, they do not allow systematic evaluation of the strategies. Two strategies may select different collaborations to the same room. If action durations were different, either strategy might perform better by chance of assigning action durations to collaborations. Unless robots are capable of approximating action durations of different collaborations, unequal action durations have no systematic impact on system performance. Since approximation of action durations highly depends on the action, we assume the general case in which robots are not capable of deducing action durations.

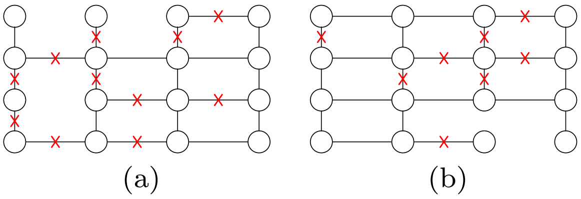

Fig. 3 Schematic presentation of randomly generated environments with 4 × 4 rooms and blocking rate **a** 0.5 and **b** 0.3. *Edges* indicate connectivity between rooms. *Red crosses* indicate blocked paths requiring collaboration

Robots may either explore, collaborate, wait, or travel. Exploration and travel time are proportional to a room's size and dimensions, respectively. Traveling robots are not assumed to simultaneously explore. Robots can only travel in rooms that are explored. Movement during exploration is half the speed while traveling to consider the map building process. At the beginning of a simulation, all robots start in the same room at the fringe of the environment. Robot interference is considered by decreasing the exploration efficiency when multiple robots are in the same room. Each simulation is performed twenty times per environment. We determine the mean values and the 0.05- and 0.95-quantiles. Individual tests with more repetitions lead to comparable results.

The robots may be coordinated in a central or distributed way. In this simulation all controllers are completely distributed. Robots share their maps and communication topology every time step if communication is possible. Global maps are constructed free of errors impacting coordination. The Fiedler value is computed by each robot based on the topology information it currently has available. Each robot tracks its frontiers. Any known room that has not been fully explored is a frontier. Frontiers are assigned by maximizing the utility value of a frontier f for robot r according to the utility function $u(r, f) = \alpha_1 d(r, f) + \alpha_2 c(r)$, which considers the approximated travel distance to reach a frontier from r's current position and the predicted connectivity c between robots at r's future location. The weights are set to $\alpha_1 = \alpha_2 = 0.5$. While they influence system behavior, the weights have no impact on the comparison of the strategies. The connectivity is approximated by the distances between robots and the well-known simple path loss model

$$\mathrm{PL}(d)\,[\mathrm{dB}] = \overline{\mathrm{PL}}(d_0) + 10\,\gamma\ \log_{10}\left(\frac{d}{d_0}\right) + X_\sigma \quad (4)$$

and parameterized according to real-world measurements in an industrial building [3]. The path loss exponent γ is set to 3.8, the reference path loss is $\overline{\mathrm{PL}}(d_0) = 45\,\mathrm{dB}$ at the reference distance $d_0 = 2\,\mathrm{m}$. The normally distributed random variable X_σ, modeling shadowing, has a variance of 10 dB. Connectivity between two robots is assumed if the received power

$$P_{\mathrm{rx}} = P_{\mathrm{tx}} - \mathrm{PL}(d)\,, \quad (5)$$

where d is the distance between robots. Finally,

$$c(r) = \begin{cases} 1 & \text{if } P_{\mathrm{rx}} \geq \Theta + \delta \\ 0 & \text{otherwise,} \end{cases} \quad (6)$$

where Θ the minimal receiver threshold and $\delta \geq 0$ a margin to improve reliability of connectivity at the new position.

Coordination is organized explicitly by tracking all other agents. Current and future locations of other robots, as far as known, are not considered as frontiers. To determine the robot to join in a collaboration, we utilize a market economy [24], but in principle any coordination algorithm may be used. The bid is computed considering travel overhead, information gain, and connectivity with equal weights.

5 Evaluation of Collaboration Strategies

5.1 General Performance Comparison

Figure 4 shows the exploration time for the four exploration strategies for two, three, and four robots and compares them to the case without collaborations required. Simulation results include 50 environments randomly generated with identical parameters. Each environment is simulated 20 times. The mean value and its 0.05- and 0.95-quantiles are computed over all $50 \cdot 20 = 1000$ experiments for each strategy. Due to the nondeterministic order of traversed frontiers and the strong dependency of a strategy's performance on the environment, the distribution of exploration time stretches over a wide range.

It can be seen that local collaborations have significant impact on exploration time, independent of the collaboration strategy and the number of robots. Compared to exploration in which all paths are freely accessible (Fig. 4 W), exploration time increases by up to 135 % for three robots and action duration of 50 time steps. Additional exploration time is due to the action time required for each collaboration and additional travel time of robots when joining a collaboration. Simulations with action duration of 0 time steps (not depicted) have an exploration time that is at least 20 % higher due to increased travel distances. The same applies for varying collaboration rates. With increasing rate, the exploration time increases.

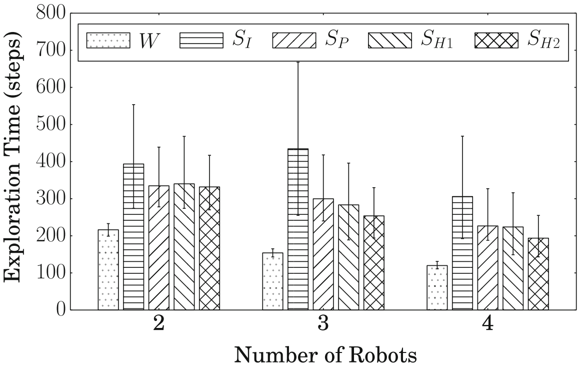

Fig. 4 Exploration time for 50 randomly generated environments

With increasing team size, the exploration times continuously decrease for all four strategies as one expects. Increasing the number of robots to six has no significant impact on mean exploration time compared to four robots. Adding additional robots slows down exploration due to robot-to-robot interference. Also, mapping errors and additional processing time for increased number of robots may slow down exploration time [4]. Hayes suggests a model to approximate team sizes and their impact on search tasks [15].

On average, S_{I} yields the longest exploration time and the maximal 0.95-quantile, independent of the number of robots. The strategies S_{P}, S_{H1}, and S_{H2} outperform S_{I} for mean exploration time and 0.95-quantile. With increasing number of robots, especially the heuristics S_{H1} and S_{H2} significantly reduce mean exploration time and its 0.05- and 0.95-quantiles. Compared to S_{I}, S_{H2} decreases the mean exploration time by approx. 40 % and 35 % for three and four robots, respectively. S_{H2} also outperforms S_{I} and S_{H1}. Mean, 0.05- and 0.95-quantiles of the exploration time are decreased by 15 % compared to S_{I}. Note that all

Table 1 Relative frequency in percent each strategy performs (a) best (minimal exploration time) and (b) worst (maximal exploration time) in 1000 randomly generated environments

(a)

	Relative frequency (%)			
Number robots	S_I	S_P	S_{H1}	S_{H2}
2	20	28	28	24
3	20	14	20	44
4	18	18	20	44

(b)

	Relative frequency (%)			
Number robots	S_I	S_P	S_{H1}	S_{H2}
2	80	6	6	8
3	98	2	0	0
4	92	6	2	0

strategies completely explore the environment having the same exploration area.

It is obvious that no single strategy may perform best in all scenarios. Environments can easily be constructed for a specific strategy to outperform the other strategies. Table 1 summarizes how often each strategy performs (a) best and (b) worst compared to the others in the 1000 experiments. For example, for two robots S_{H2} yielded the shortest (best) exploration time in 24 % of the randomly generated environments, the longest (worst) exploration time in 8 %.

While for two robots all strategies perform approximately equally often best, for three and four robots S_{H2} dominates in almost half of the cases. In cases where S_{H2} does not perform best, it still performs comparable to the other strategies yielding a minimal 0.05-quantile. Only for two robots S_{H2} performs worst in 8 % of the environments, for three and four robots it never performs worse than the other strategies. In comparison, while S_{I} outperforms the other strategies in 20 % of the cases, the mean exploration time and the 0.95-quantile are high. In other words, if S_{I} does not perform best, it will perform much worse than the others. In 80 % and more, S_{I} performs worst.

Both heuristics decrease the mean exploration time and its 0.05- and 0.95-quantiles. They perform best more often than the two obvious strategies and less often worse. S_{H2} yields shortest exploration time and performs worst least.

5.2 Heuristic 1

We elaborate the behavior of S_{H1} in more detail. Figure 5 shows the exploration time for the environment depicted in Fig. 3a. The high exploration time for S_{I} is due to a higher number of collaborations. Figure 6 shows the mean number of collaborations performed during the 20 simulations. While the minimum number of collaborations required to fully explore the environment is six, S_{I} leads to eight partly redundant collaborations extending the exploration time. In comparison S_{P} and S_{H1} aim to reduce the number of collaborations at the expense of additional travel overhead.

In the example of the environment in Fig. 3a, both S_{P} and S_{H1} continue to explore until only blocked paths are available. While robots in S_{P} aim for the closest collaboration, S_{H1} selects the collaboration with the least number of alternative routes; assuming the more alternative routes available, the higher the likelihood of not having to collaborate. While it decreases the probability of redundant collaborations, it increases travel distance as depicted in Fig. 7. With increasing collaboration duration, the time taken for traveling becomes neglectable compared to the collaboration time, and S_{H2} outperforms S_{P}.

So far we assumed the reference map for S_{H1} to be identical with the yet unknown environment.

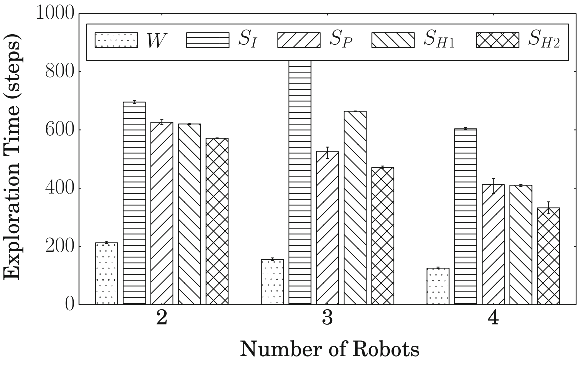

Fig. 5 Exploration time in the environment depicted in Fig. 3a

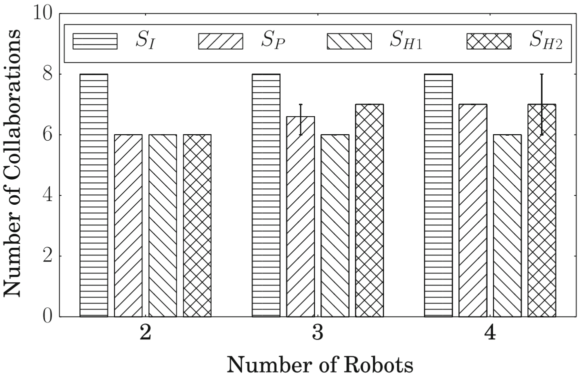

Fig. 6 Total number of collaborations performed for the environment depicted in Fig. 3a

Derivations between the reference map and the actual floor plan not changing the number of routes between rooms have no impact. However, wrong information on the degree of a vertex may lead to immediate, unnecessary collaborations or collaborations may be postponed though no alternative path is actually available. For example, using the environment depicted in Fig. 3b as a reference map for the environment in Fig. 3a increases the mean exploration time by 15 % for two, three, and four robots due to an additional, unnecessary collaboration.

5.3 Heuristic 2

Figures 8 and 9 indicate exploration time and number of collaborations, respectively, for the environment depicted in Fig. 3b. For two and three, robots S_P, S_{H1}, and S_{H2} require the same number of collaborations; for four robots S_{H1} requires up to three. Despite the same number of collaborations, however, S_{H2} outperforms S_{H1} and S_P for three and four robots. The same can be seen in the previous example (Fig. 5).

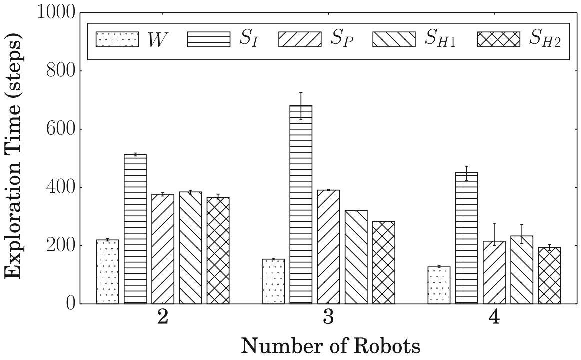

Fig. 8 Exploration time for the environment depicted in Fig. 3b

This improvement can be explained by the time steps when explorations occur. Figure 10 shows the average time steps when the first and second collaborations take place. For example, the first collaboration for S_I and two robots happens on average at 83 time steps. The second collaboration after 173 time steps. It can be seen that for S_{H1} the first collaborations occur earlier than for the other strategies, for two robots 75 time steps or 28 % earlier. Clearing paths allows to push forward into regions which cannot be reached otherwise. This increases the number of frontiers from which agents can select. Increasing the number of frontiers increases the performance of the utility function which selects the most suitable frontier from the currently known ones — especially if the

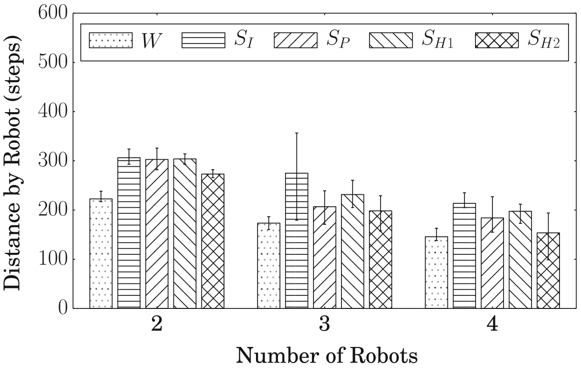

Fig. 7 Travel distance by robot for the environment depicted in Fig. 3a

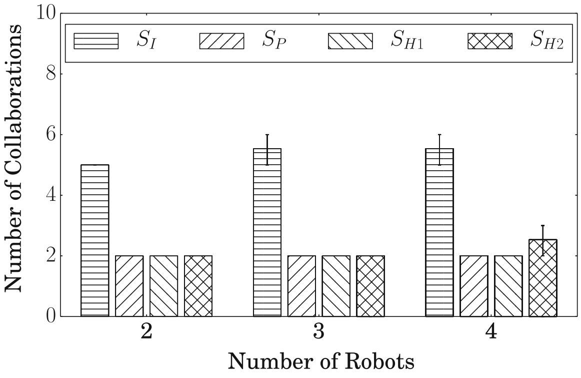

Fig. 9 Total number of collaborations performed for the environment depicted in Fig. 3b

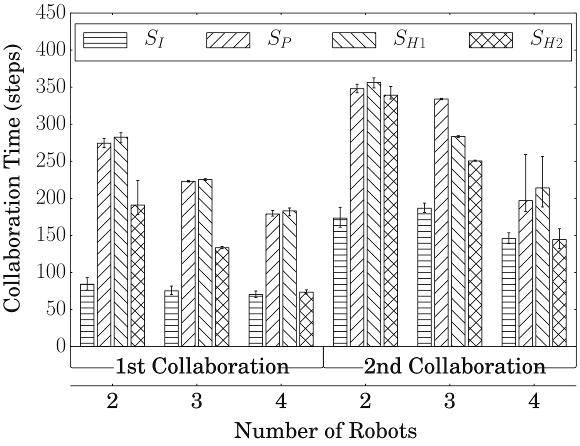

Fig. 10 Mean time steps t_{coll} when the first and second collaborative action occurred (environment Fig. 3b)

number of frontiers falls below the number of robots. We assume robots operate best without interfering with each other. If collaborating increases the number of frontiers above the number of robots, frontiers can be assigned more efficiently speeding up exploration. Collaborations happen much earlier for S_{I} than for the other two strategies. These early collaborations turn out to be unnecessary and do not improve frontier assignment to make up for the collaboration overhead.

Otherwise the strategies show similar performance in exploration time compared to Figs. 4 and 5 with the exception of S_{I} and three robots taking significantly longer than for two robots. The reason for S_{I} to slow down exploration is twofold. Firstly, for three robots the average number of collaborations is 5.5 compared to 5.0 for two robots. This leads to longer exploration times. Secondly, we assume two robots are required for collaborative actions for which three is a bad match. If the third robot requires a local collaboration while the other two robots are already collaborating, it has to wait until another robot is available. While waiting the robot does not continue to explore.

The average cumulative travel distance depicted in Fig. 11 supports these results of better frontier selection. For example, all four robots jointly traveled on average 747 steps in case of S_{I}. For three and four robots the travel distance for the strategy S_{H2} is decreased compared to the other strategies due to not having to track back to previously visited collaboration points and improved frontier assignment by the utility function.

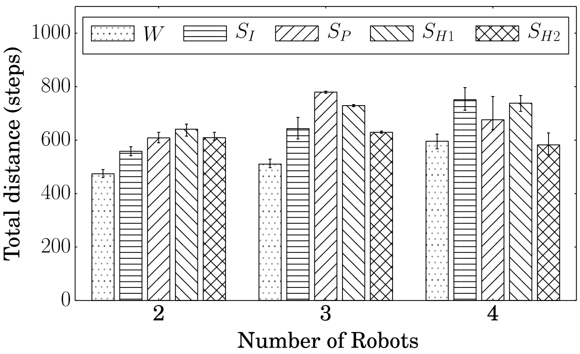

Fig. 11 Cumulative travel distance for the environment depicted in Fig. 3b

The total travel distance increases with increasing number of robots. Paths which have to be followed by all robots because alternative paths are not available contribute multiple times to the total distance. This is a general trade-off in multi-robot systems.

6 Conclusions

Multi-robot exploration with local collaboration requires robots to physically join another robot at its location to assist in a collaborative task. We classified collaborative actions and derived heuristics to judge the necessity of collaboration with limited knowledge of the environment.

Considering collaborations that become mandatory under certain circumstances, e.g., when having to clear a path to be able to explore a blocked area, four strategies were compared that decide as to whether to collaborate and if so when. All strategies allow fully autonomous exploration of indoor environments. Their efficiencies significantly depend on the layout of a building having trade-offs with respect to number of required collaborations and travel distance. In randomly generated environments, however, it shows that the strategies using the heuristics reduce exploration time benefiting from building patterns learned during an exploration. The strategies including the heuristics also decrease travel distance which is important for resource limited systems.

Acknowledgments This work was partly supported by the ERDF KWF, and BABEG under grant KWF-20214/24272/36084 (SINUS). It was performed in the research cluster Lakeside Labs.

References

1. Alexander, C., Ishikawa, S., Silverstein, M.: A Pattern Language. Oxford University Press (1977)
2. Andre, T., Bettstetter, C.: Assessing the value of coordination in mobile robot exploration using a discrete-time Markov process. In: Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (2013)
3. Andre, T., Brandner, G., Marchenko, N., Bettstetter, C.: Measurement-based analysis of cooperative relaying in an industrial wireless sensor network. In: Proceedings of IEEE GLOBECOM (2012)
4. Andre, T., Neuhold, D., Bettstetter, C.: Coordinated multi-robot exploration: Out of the box packages for ROS. In: Proceedings of IEEE GLOBECOM WiUAV Workshop (2014)
5. Awerbuch, B., Betke, M., Rivest, R.L., Singh, M.: Piecemeal graph exploration by a mobile robot. Inf. Comput. **152**(2), 155–172 (1999)
6. Burgard, W., Moors, M., Schneider, F.: Collaborative exploration of unknown environments with teams of mobile robots. In: Beetz, M., Hertzberg, J., Ghallab, M., Pollack, M. (eds.) Advances in Plan-Based Control of Robotic Agents, Lecture Notes in Computer Science, vol. 2466, pp. 52–70. Springer, Berlin (2002)
7. Burgard, W., Moors, M., Stachniss, C., Schneider, F.E.: Coordinated multi-robot exploration. IEEE Trans. Robot. **21**, 376–386 (2005)
8. Bürger, G.: Personal communication (2014)
9. Dorigo, M. et al.: Swarmanoid: A novel concept for the study of heterogeneous robotic swarms. IEEE Robot. Autom. Mag. **20**(4), 60–71 (2013)
10. Espinace, P., Kollar, T., Roy, N., Soto, A.: Indoor scene recognition by a mobile robot through adaptive object detection. Robot. Auton. Syst. **61**(9), 932–947 (2013)
11. Espino, J.C., Steux, B., Hamzaoui, O.E.: Safe navigating system for indoor environments. In: Proceedings of 5th International Conference on Automation, Robotics and Applications (ICARA) (2011)
12. Farinelli, A., Iocchi, L., Nardi, D.: Multirobot systems: a classification focused on coordination. IEEE Trans. Syst., Man, Cybern. **34**(5), 2015–2028 (2004)
13. Godsil, C., Royle, G.: Algebraic Graph Theory. Springer (2001)
14. González-Baños, H.H., Latombe, J.C.: Navigation strategies for exploring indoor environments. Int. J. Robot. Res. **21**(10-11), 829–848 (2002)
15. Hayes, A.: How many robots? Group size and efficiency in collective search tasks. In: Asama, H., Arai, T., Fukuda, T., Hasegawa, T. (eds.) Distributed Autonomous Robotic Systems 5, pp. 289–298. Springer, Japan (2002)
16. de Hoog, J., Cameron, S., Visser, A.: Selection of rendezvous points for multi-robot exploration in dynamic environments. In: Proceedings of International Conference on Auton. Agents and Multi-Agent Systems (AAMAS) (2010)
17. Juliá, M., Gil, A., Reinoso, O.: A comparison of path planning strategies for autonomous exploration and mapping of unknown environments. Auton. Robot. **33**, 427–444 (2012)
18. Michael, N., Fink, J., Kumar, V.: Cooperative manipulation and transportation with aerial robots. Auton. Robot. **30**, 73–86 (2011)
19. Mostofi, Y.: Communication-aware motion planning in fading environments. In: Proceedings of IEEE International Conference Robotics and Automation (ICRA) (2008)
20. Nevatia, Y., et al.: Augmented autonomy: Improving human-robot team performance in urban search and rescue. In: Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (2008)
21. Pei, Y., Mutka, M.W.: Joint bandwidth-aware relay placement and routing in heterogeneous wireless networks. In: Proceedings of IEEE International Conference on Parallel and Distributed Systems (ICPADS) (2011)
22. Singh, K., Fujimura, K.: Map making by cooperating mobile robots. In: Proceedings of IEEE International Conference on Robotics and Automation (ICRA) (1993)
23. Wurm, K.M., Dornhege, C., Eyerich, P., Stachniss, C., Nebel, B., Burgard, W.: Coordinated exploration with marsupial teams of robots using temporal symbolic planning. In: Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (2010)
24. Zlot, R.M., Stentz, A., Dias, M., Thayer, S.: Multi-robot exploration controlled by a market economy. In: Proceedings of IEEE International Conference on Robotics and Automation (ICRA) (2002)

Torsten Andre (S'11) is a researcher and Ph.D. student at the Institute of Networked and Embedded Systems at Alpen-Adria-Universität Klagenfurt, Austria. He received his diploma from RWTH Aachen, Germany, in 2010. His research focuses on the coordination of multi-robot systems in indoor environments allowing collaborative tasks. Related to coordination, communication in multi-robot and ad-hoc networks is a fundamental part of his work. He serves as Executive Member in the IEEE Austria Section since 2014 and is founding member of the IEEE Student Branch Klagenfurt.

Christian Bettstetter (S'98-M'04-SM'09) received the Dipl.-Ing. degree in 1998 and the Dr.-Ing. degree (summa cum laude) in 2004, both in electrical engineering and information technology from the Technische Universität München (TUM), Munich, Germany. He was a Staff Member with the Communications Networks Institute, TUM, until 2003. From 2003 to 2005, he was a Senior Researcher with the DOCOMO Euro-Labs. Since 2005, he has been a Professor and Head of the Institute of Networked and Embedded Systems, University of Klagenfurt, Austria. He is also Scientific Director and Founder of Lakeside Labs, Klagenfurt, a research cluster on self-organizing networked systems. He coauthored the textbook GSM-Architecture, Protocols and Services (Wiley). Mr. Bettstetter received Best Paper Awards from the IEEE Vehicular Technology Society and the German Information Technology Society (ITG).