Search and Rescue with Sparsely Connected Swarms

Ulrich Dah-Achinanon[1*†], Seyed Ehsan Marjani Bajestani[1*†], Pierre-Yves Lajoie[1†] and Giovanni Beltrame[1]

[1*]MIST Lab., Department of computer engineering and software engineering, Polytechnique Montréal, Chemin de Polytechnique, Montréal, H3T 1J4, Québec, Canada.

*Corresponding author(s). E-mail(s): ulrich.dah-achinanon@polymtl.ca; ehsan.marjani@polymtl.ca;
Contributing authors: pierre-yves.lajoie@polymtl.ca; giovanni.beltrame@polymtl.ca;
†These authors contributed equally to this work.

Abstract

Designing and deploying autonomous swarms capable of performing collective tasks in real-world is extremely challenging. One drawback of getting out of the lab is that realistic tasks involve long distances with limited numbers of robots, leading to sparse and intermittent connectivity. As an example, search and rescue (SAR) requires robots to coordinate in their search, and relay the information of found targets. The search's effectiveness is greatly reduced if robots must stay close to maintain connectivity. This paper proposes a decentralized search system that only requires sporadic connectivity and allows information diffusion through the swarm whenever possible. Our robots share and update a distributed belief map, to coordinate the search. Once a target is detected, the robots form a communication relay between a base station and the target's position. We show the applicability of our system both in simulation and with real-world experiments with a small swarm of drones.

Keywords: Swarm robotics, Drone, Sparse connection, Search-And-Rescue, Real-world deployment, Rendezvous point

1 Introduction

Drones or unmanned aerial vehicules (UAVs) have been experiencing steady growth for the past few years in terms of their popularity, availability, and potential. This enables a wide range of applications in several fields going from surveillance to search and rescue missions (Apvrille et al, 2014). The emergence of swarms systems, made of multiple UAVs collaborating towards a common goal, further improves the efficiency of those solutions. Swarm robotics has been defined by Brambilla et al (2013), as an approach to collective robotics inspired by the self-organized behaviors of social

Fig. 1 The real-world experiments setup with three DJI M300 quad-copters

animals, where a large group of simple robots aims to accomplish a complex task through simple rules and local interactions.

Using teams of robots, instead of a single one, brings robustness, scalability, survivability, and it increases the speed of execution (Hentati and Fourati, 2020). Despite those practical benefits, many unresolved challenges prevent swarm robotics from being used in commercial products. In particular, swarms of UAVs are confronted with multiple challenges such as: in-flight coordination, swarm layout reconfiguration, handling losses of swarms elements, and data relaying optimization among others (Wubben et al, 2020). Tarapore et al (2020) outline one of those challenges and formalizes the notion of sparse swarms in which it can be prohibitively expensive for the robots to maintain close proximity. For example, during swarm-based search and rescue (SAR) operations, preserving close proximity among the robots would certainly restrain the area that can be covered. Therefore, a robust ad-hoc communication system, resilient to disconnections between the swarm agents, is essential to deploy such systems in realistic scenarios.

In this article, we attempt to bridge the gap between theoretical approaches and practical applications by proposing a SAR algorithm based on ad-hoc networks accepting sporadic connectivity. This algorithm leverages a search pattern accepting sparsely connected swarms by scheduling a rendezvous point for periodic meetings and target discovery report. It then allows the swarm to adopt a relay tree formation connecting the meeting point (or base) to all the detected targets. Thus, the connectivity between the ground operators and the robots is restored only when targets are found. The search method is based on belief space exploration in order to incorporate crucial priors from the authorities, such as the last known locations of the targets.

This paper argues that ad-hoc networks accepting sporadic connectivity are the key to real-world deployments of swarms of UAVs in large areas. Such networks should handle adaptive topology and routing while assuring reliable data exchange within the swarm. Existing works exploit either a rendezvous point or a relay chain formation to improve the communication links, but, to the best of our knowledge, none combine these for SAR applications. For instance, Nickerson (2004), Hourani et al (2013), and Belkadi et al (2016) leverage a rendezvous point for exploration and SAR, but they do not use communication relays. Other works present relay chain architectures for communication enhancement (Li, 2019; Varadharajan et al, 2020), without considering the search pattern. This article aims to bridge these two principles with the mentioned algorithm. To summarize, the main contributions presented in this paper are:

- a search pattern based on rendezvous point, using ad-hoc networks with adaptive topology and routing, and accepting sparsely connected swarms;
- an algorithm to create an adaptive relay tree structure design to maintain the communication between the ground operators and the discovered targets;
- an implementation of a search method based on dynamic, distributed belief maps;
- an experimental robotic system for real-world deployment of swarms of drones;
- simulation and real-world deployments of the proposed system.

We validate our approach through tests in simulation and real-world experiments. In simulation, we test our dynamic belief map search algorithm on different area sizes and number of drones. We also performed real-world field tests with three drones to confirm our findings. Figure 1 shows the real-world experiments setup with three DJI M300 quad-copters.

The rest of the paper is organized as follows: in section 2 we present related works outlining similarities and differences with our approach. In Section 3 we describe our algorithm and its components. Sections 4 and 5 present our simulations and experiments setup and results. Finally, Section 6 draws concluding remarks while presenting possible future works.

2 Related work

As mentioned by Hentati and Fourati (2020), a reliable communication structure is essential to share information among group of neighbors in swarm applications. Sharing the information becomes essential in applications, such as SAR, where a group of robots is trying to find a target in a large area. As indicated by Alotaibi et al (2019), finding the target could be faster using more robots, however, the lack of reliable communication could separate a group of robots or make the whole mission fail, especially in centralized applications. For instance, Alotaibi et al (2019) proposed a layered search and rescue (LSAR) centralized “partitioning” algorithm, that needs reliable communication with a cloud server. Although their simulation results indicate that a better success rate could be achieved by increasing the number of robots, it is inherently limited by the central communication bottleneck to the server. Recent work (Ruetten et al, 2020) introduced an optimized self-organized mesh network to cover large areas, but do not consider disconnections.

To solve these communication issues in a team of robots during exploration missions, Hourani et al (2013) considered a periodic rendezvous strategy in order to overlap the communication ranges of the robots. It also presented an approach to mitigate the negative impact of these meetings on the time efficiency of the overall mission. Our approach is similar, but we drop the connectivity maintenance requirement during the searching phase. Another benefit of our technique is that the drones continue to search for the targets while going to the periodic meetings. Andries and Charpillet (2013) and Andries and Charpillet (2015) also used a meeting point, but the robots only meet at the rendezvous when the exploration is completed. Adopting a different approach, Belkadi et al (2016) plans the exploration in order to converge to a predefined spatial configuration around the rendezvous point.

Belief maps have been studied for a long time as a tool for multi-robot exploration (Kobayashi et al, 2002, 2003). Similar to our work, Khan et al (2014) updates the belief map with local observations and merges data from multiple UAVs. Our distributed belief map implementation is an adaptation of the work proposed by Vielfaure et al (2021), in which the authors stored the shared belief map in a distributed database called virtual stigmergy (Pinciroli et al, 2016).

During the rescue phase of SAR missions, it is crucial to maintain the connection between the base station and the UAVs following the target. To this end, we propose to maintain a relay chain from the rendezvous position to the targets once these latter are found. Using a heuristic optimization method, Kim et al (2020) increased the communication performance metric and determined the optimal positions for the communication relay robots. To keep the connection between a heterogeneous group of robots (on-ground and flying robots), Varadharajan et al (2020) introduced a fully decentralized algorithm to create and keep a chain of robots from the ground station to the target. Li (2019) and Zhang et al (2021) also used drones as relays and virtual potentials to create a stable link between a group of robots (or a survey drone) and a base station. Instead of using virtual potentials, Yamaguchi et al (2017) measured the communication quality and expanded the drone relays when needed. While the abovementioned works focused on the formation of relay chains during the search phase, this could make the search process very slow, which would be critical for SAR operations. Therefore, we propose to create the communication relay chains only in the rescue phase. Our communication relay approach is similar to the approach proposed by Majcherczyk et al (2018); Çeltek et al (2018), in which the creation and expansion of tree/chain topologies between drones and target(s) have been evaluated. We use a tree topology for the relay connection when rescuing more than one target.

3 Search and Rescue with Sparsely Connected Swarms

In this paper, we consider the scenario in which a swarm of drones needs to be deployed in an unknown environment to search for one or more targets, and track them as rescuers are dispatched to the target locations. The drones explore the area autonomously and in a decentralized manner, searching for targets. Communication links are needed between the swarm members either to inform the others when a target is found and to share the target positions, propagating this information to a base station so that the targets can be rescued.

In realistic scenarios, the search area is likely to be larger than the combined communication coverage of the robots in the swarm. Therefore, the searching robots need to disconnect from their neighbors to explore enough space to find the desired targets, creating a sparse swarm.

Let us consider a swarm $\boldsymbol{S}$ of n robots $\boldsymbol{S} = \{1, 2, ..., n\}$. At a given time step $t_s \geq 0$ during the mission, $\boldsymbol{S}$ is considered a sparse swarm if a robot $r \in \boldsymbol{S}$ satisfies:

$$cost_r(\text{"}move \quad to \quad nearest \quad neighbor\text{"}, t_s) \gg cost(\text{"}perform \quad typical \quad operation\text{"}, t_s) \tag{1}$$

where $\gg$ is defined as "at least one order of magnitude greater than" and $cost_r$ is a function defining the cost for robot r to perform a given task at a given time (Tarapore et al, 2020). In order to ensure coordination and efficient searching in such a swarm, we designed an algorithm inspired from typical search parties in rescue operations, shown in Figure 2.

The overall idea is that robots perform their search for a target, regularly reporting at a fixed rendezvous location or meeting point. If a robot finds a target, it immediately goes to the meeting point to share the location of the target with the operator and the rest of the swarm. Assuming a robot is the first to go to the meeting point after finding a target, it becomes a *root* robot, and it coordinates the formation of a relay chain towards the target. The chain construction starts as the root robot broadcasts a call for *networkers* (i.e., relay robots) that other members of the swarm respond to with bids based on their distance from the required position of the relay, and the root robot assigns roles based on the received bids(Gerkey and Mataric, 2002). As robots find more targets, the root adds branches to the relay formation, and should robots find targets simultaneously, they elect a root through a basic consensus mechanism.

3.1 Agent roles

The overall strategy is based on a state machine that assigns roles to the robots in the swarm, with each role is associated with a task executed by the robot.

Searcher: when a robot is a searcher, it looks for targets using a predefined search method (a belief-based search in our case). During its operation, a searcher listens and responds to *calls for bids* from other agents. These calls offer networker (i.e., relay) roles, and a searcher bids based on its distance to the requested relay position.

Root candidate: upon finding a new target, an agent will go to the rendezvous point to become a root robot. In case of a root being already present, the agent shares its target information with the existing root and go back to being a searcher. If multiple robots are heading to the rendezvous point, the first robot to arrive proclaims itself the winner and shares that information with every incoming robots. Should multiple agents arrive to the rendezvous at the same time, we use a conflict management method based on robot ID (Pinciroli et al, 2016).

Root: winning the bid for the root node, an agent becomes the root: it listens for new target information from the swarm, it computes the number of networkers needed per target and their positions, and calls for robots to fill the networker roles. Note that this strategy does not make the system centralized: the root is easily replaced in case of failure with a new election.

Auctioneer: when the root needs networkers to cover a target, it switches to the auctioneer state, for a typical market-based task allocation strategy (Gerkey and Mataric, 2002). The root/auctioneer broadcasts the relay position, opens the auction, listens for bids, and closes the auction after a predefined period of time, remaining in the same state until a winner is found. The auctioneer then broadcasts the winning bid and goes back to the root state.

Networker bidder: when a searcher receives a call for bids, it stops moving and bids for a networker role. Its bid value is inversely proportional to its distance to the assigned relay position.

Networker: the networker bidder that is the closest to a relay position wins the bid and becomes a networker. A network relays information between a target location and a base station at the meeting point, connecting the target with the operator and providing constant communication coverage in the area of the target.

Rendezvous: robots regularly switch to the rendezvous state and go back to the meeting point to check for any new information (new root node, request for a networker, or updated found targets list). Note that robots keep searching for targets on their way back to the meeting point. If there are no networker calls for bids happening, the robots in rendezvous state go directly back to being searchers.

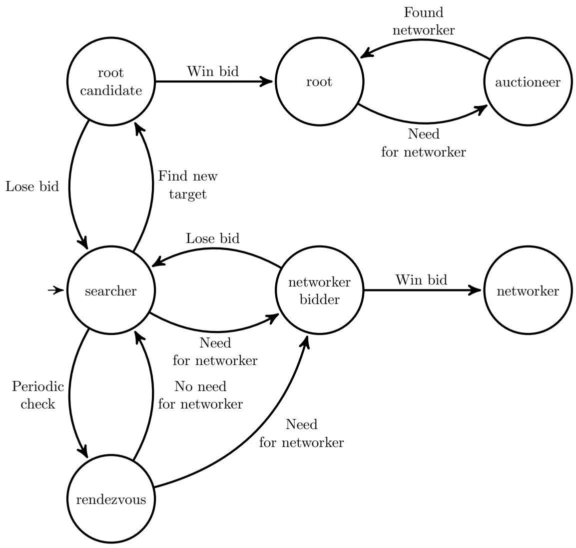

Fig. 2 Coordination algorithm for target searching

3.2 Algorithm

All the robots execute their search for the targets based on the available belief information. The belief information, represented as a map, is updated and distributed to the neighbors during the search to avoid searching the same area multiple times. This distributed belief map based search is inspired from the work in Vielfaure et al (2021). To distribute a belief map, we use the virtual stigmergy (VS), a system that allows a swarm of robots to agree on a set of (key,value) pairs through gossip communication (Pinciroli et al, 2016). Thus, at each step, a searcher will decrease the belief value for its current position if no target is detected there. The new value is then put into the VS, sharing the information with all neighbors in communication range. A searcher sampling a new position to navigate to, will then opportunistically get the most recent belief for the position (from the initial map if no updated version is available in the VS). It is worth noting that the propagation of the VS is strictly best-effort, and therefore tolerant to disconnections and communication delays. Based on the value of the belief map, the robots decide to either move to the sampled position or sample a new one (if the belief is below a certain threshold).

The idea of the search pattern is to realize multiple runs of a user-defined duration and come back to a rendezvous point between each of them. During the search, if a robot finds a target, it goes back to the initially fixed rally point and checks for the existence of a root node. The first drone to come back to the rally point after finding a target will be the root. When multiple drones arrive to the rendezvous at the same time, a conflict management routine selects one of the drones as the root. This robot becomes the first link of the communication and tracking relay between the meeting point and the targets. The root stays at the rendezvous point and broadcasts relevant information (updated found targets list, root id, networking positions, etc.) to all the robots in its communication range. As explained previously, this node computes the networkers' positions and manages an auction every time it needs a new networker by switching to the auctioneer state. The networkers's positions depend on the *communication range* of the robots. Let N be the set of networkers positions in the system and $\boldsymbol{r}$ the root's position. To find the networkers positions for a target located at $\boldsymbol{t}$, we choose a branching node at a position $\boldsymbol{b}$ such that:

$$\|\boldsymbol{t}-\boldsymbol{b}\| = min(\|\boldsymbol{t}-\boldsymbol{n}\|) \tag{2}$$

for all $\boldsymbol{n} \in N \bigcup r$.

The networkers are then placed one after the other on the line connecting $\boldsymbol{b}$ and $\boldsymbol{t}$. They are spaced at a maximum distance of *communication range* to ensure connectivity in the relay.

Other robots that did not find any target, go back periodically to the meeting point to check if another robot found a target or if a networker is needed. Since the battery life of flying robots is quite limited, this periodic check is an opportunity for recharging or battery swapping. When reaching the meeting point, if the robot receives a message from the root for a networking position, it immediately sends its bid for the auction and waits for the results announcement. When a robot wins the bid it acknowledges the auctioneer that it received the message and goes to its assigned position, becoming part of the communication and tracking relay. The robots in relay positions form a tree from the meeting point, allowing the operator at a base station to constantly and simultaneously to “see” and monitor the state of all the detected targets.

With a sufficient number of searcher robots, the relay should allow connectivity maintenance and a live camera stream of the target to the ground operators. In the case of a moving target, the closest robot to target has the responsibility of tracking its motion and sharing the updated position along the relay. This way, the relay can adapt itself and follow the target up until its rescue.

3.3 Real-world deployment

For the real-world deployment of our solution, we used DJI Matrice 300 RTK (M300 RTK) drones equipped with a Manifold 2 onboard computer which was connected to the drone through a serial connection. The M300 RTK is a powerful UAV platform offering an adaptive onboard software development kit (OSDK) for autonomous control of the aircraft. It uses an advanced flight controller system, a 6 directional sensing and positioning system and FPV camera. Thanks to these features, the drones were able to perform a basic collision avoidance routine during the flights.

The decentralized control of the drones is achieved using Buzz, a domain-specific language designed for programming multi-robot teams and swarms behaviors (Pinciroli and Beltrame, 2016). The software consists of three main layers: the Buzz control layer taking care of the algorithm logic, the ROSBuzz (St-Onge et al, 2017) layer responsible for the integration of the swarm-oriented programming language and its virtual machine (BVM) into the ROS environment, and the DJI OSDK layer that manages the flight controller and other UAV related features. The Buzz control layer is responsible for the system's behavior, using a Buzz script to implement the proposed algorithm, while sending hardware specific commands to the lower layers.

Our whole experimental system is based on ROS (Robot Operating System), an open-source and now standard software system for robotic development (Quigley et al, 2009). To link the Buzz control layer to ROS, we use ROSBuzz, an existing implementation of the BVM as a ROS node. ROSBuzz encapsulates all the BVM logic, publishes the Buzz script commands, and subscribes to external data such as sensor readings. A main feature of Buzz is the implementation of gossip-based situated communication among neighbors in a swarm. This feature is implemented in our system with batman-adv (Better Approach to Mobile Ad-hoc Networking), using a ROS node that manages the neighbors of each robots and broadcasts messages as needed by the Buzz script. Batman-adv is a layer 2-based protocol leveraging adaptive topology and routing to offer a robust ad-hoc networking solution (Kiran et al, 2018). As our algorithm assumes sporadic connectivity among the robots in the swarm, batman-adv can easily handle such a pattern, ensuring a reliable link between the robots when in communication range. The physical device supporting batman-adv is the 5 GHz WiFi antenna on the Manifold 2, set to communicate on an common IBSS ad-hoc wireless network.

The DJI OSDK layer is a DJI proprietary API that allows a developer to control the aircraft with a program. DJI proposes a version of the OSDK integrate with ROS that we used in our setup, and the UAVs are actuated via service calls. The interaction between the ROSBuzz ecosystem and the OSDK layer is taken care of by a custom adapter node. This node receives actuation commands from the Buzz script (in the form of topics) and sends flight controller-specific commands to the OSDK, as well as publishing the required data for ROSBuzz's operation (e.g., GNSS readings). Such an architecture, allows for a easily portable code base. In fact, since the adapter is the only M300 dependent node, using the same algorithm with a different robotic platform only requires to write a similar adapter. Figure 3 synthesizes the described software architecture, which is completely open source (see github.com/mistlab).

4 Experimental results: simulations

To validate the presented algorithm and evaluate the necessary time required to find targets and obtain the final relay chain, we performed a series of tests in a simulated environment. We used 15, 20, and 25 robots while randomly varying their starting positions and the position of the targets. We also compare the performance of a random walk search to our dynamic belief map search, while using in both cases the presented search pattern.

4.1 Simulation setup

We use ARGoS, a multi-physics robot simulator (Pinciroli et al, 2012). The experiments were performed on an ARGoS model of the Spiri Mu

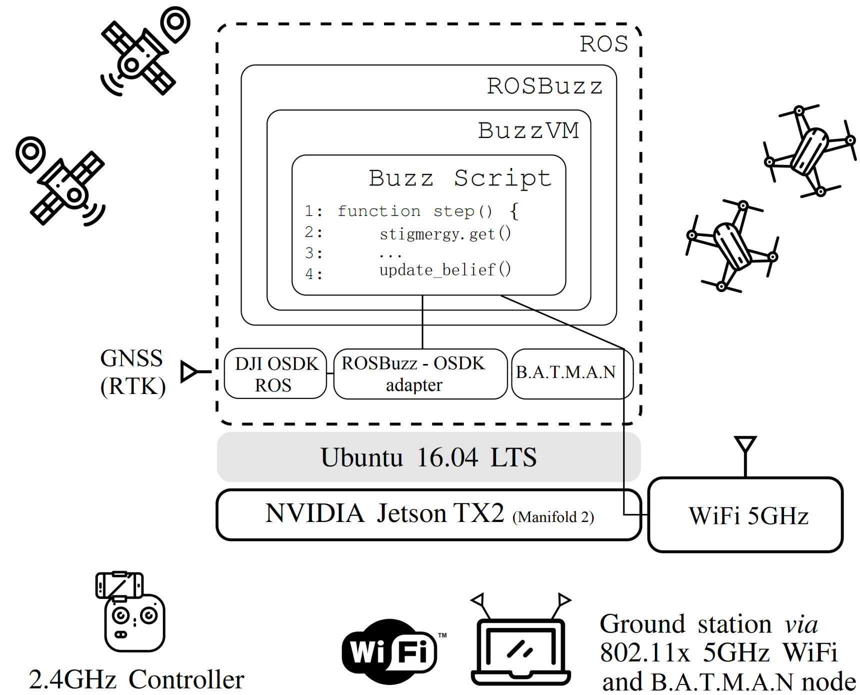

Fig. 3 The control architecture. The global positioning system and (back-up) remote controllers joint with the DJIs OSDK and flight controller to perform the decentralized behavioural Buzz script. The entire fleet runs the same script, interfacing with the Flight Control Unit (FCU) through DJI OSDK ROS and communication device (WiFi) through the Robot Operating System (ROS). The communication between swarm has been achieved by creating a B.A.T.M.A.N ad-hoc mesh network and using the DJI-Manifold 2 WiFi as the network hardware (St-Onge et al, 2019)

quadrotor from Spiri Robotics. Three foot-bot mobile robots (Dorigo et al, 2013) were randomly spawned and used as targets to be detected by the Spiris. To perform the detection, we simulate a basic sensing mechanism on the drones with a downward facing camera using blob detection. Figure 4 presents an example of a starting state of the simulation.

We first used a random walk exploration pattern (Dimidov et al, 2016) where each drone sequentially samples a 2D position and autonomously navigates to it(keeping the height constant). The second search method was is the proposed dynamic belief space exploration. In both cases, we simulate the system until all the targets are detected and the relay network is fully formed. The total time needed to find the targets is the metric used for our evaluation.

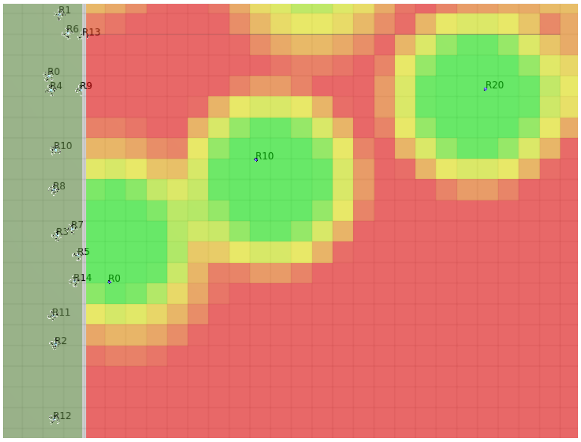

Fig. 4 Initial simulation setup sample using 15 drones (R0 to R19 on the left side) with 3 targets (R0, R10, and R20 in the searching area). The searching area is represented as a belief 2D map where red represent a probability close to 0 to find the target and green represent a probability close to 1

4.2 Results

We perform six test cases with three arena sizes: *(i)* 15 drones with random search, *(ii)* 20 drones with random search, *(iii)* 25 drones with random search, *(iv)* 15 drones with belief space search, *(v)* 20 drones with belief space search, and *(vi)* 25 drones with belief space search. We used three arena sizes: *(a)* 20m x 20m, **(b)** 30m x 30m, and **(c)** 40m x 40m. To obtain statistically relevant results, each test case was executed 30 times, randomly assigning the initial positions for the drones, the targets' positions and the belief map if applicable. The other parameters, including the meeting point, were maintained constant through the experiments: 20 search steps, three targets and a communication range of of 10 meters (not realistic, but distances can be simply scaled to realistic values). An output sample for a test with with 15 drones and 3 targets is presented in Figure 5 where we can see the relay tree connecting the meeting point to the three targets. The figure also show the remaining drones (not used in the relay) searching for any additional target.

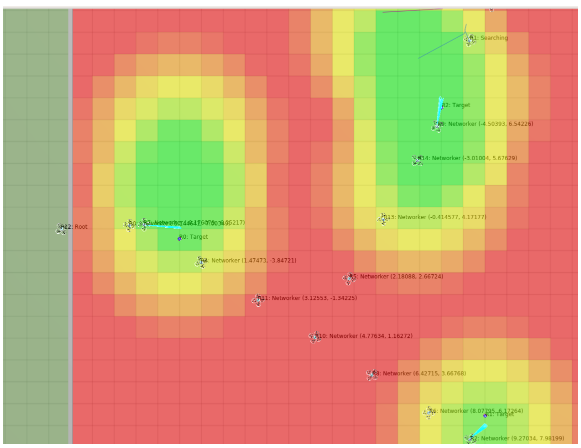

Fig. 5 Possible final state for a test with 15 drones and 3 targets and a communication range of 4 meters. Each drone logging its state: root, networker or serching. The searching area is represented as a belief 2D map where red represent a probability close to 0 to find the target and green represent a probability close to 1

For every experiment, we report the number of Buzz timesteps necessary to find the three targets. For reference, a Buzz timestep is 0.1s by default but it is fully configurable depending on the capabilities of the robots and communication system. Figures 6, 7 and 8 present the number of timesteps necessary to find the targets for different arena sizes and different number of drones. They show the results obtained with both the random search and the belief space search while increasing the number of drones and the arena size. From those results, we can see as expected for a swarm based SAR algorithm that the time to find the target decreases when the number of drones increases. This observation is due to the fact that a bigger search area can be covered with more drones, allowing them to find the targets faster.

Also, the time decreases in all cases when we use a belief space search in comparison to a random walk search. In fact, by reducing step after step the searching area (thanks to the distributed belief information), our search method allows drones to converge faster towards the targets.

Finally, when the arena size increases we can see that the time needed to find the targets increases as well (the sub graphs have different vertical scales). This can be explained by the sampling space which gets bigger with the arena. Thus, there is more options to explore before we can find the target wherever it might be.

To confirm the hypothesis that the belief space search would be faster than the random walk, in any case, we did a Bayesian paired samples t-test on the results by using JASP Team (2021) software. The results show that all data accept the hypothesis with the Bayesian Factor (BF) greater than 1 (BF min = 2.886, BF max = 66550.284).

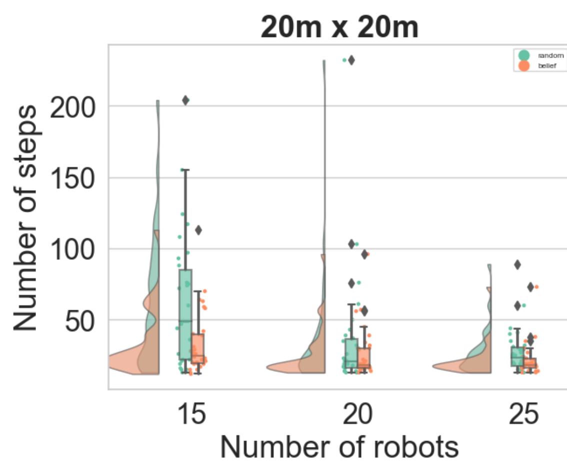

Fig. 6 Number of timesteps to find 3 targets in a 20 m x 20 m arena. The results for the random search are in green and the results for the belief space search are in orange. The BF and the error of the Bayesian paired samples t-test for each setup are respectively 92.404, and 8.273e-5 for 15 robots, 2.886 and 0.001 for 20 robots, and 70.606, and 1.043e-4 for 25 robots.

Another hypothesis we postulated was that the search pattern that was designed should create the relay formation in nearly constant time. To verify that assertion, we considered for each experiment

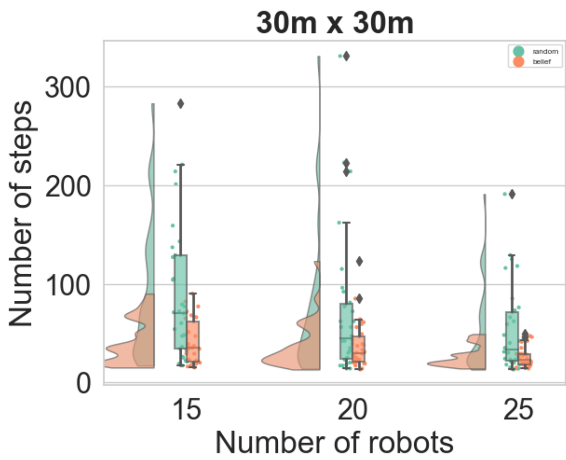

Fig. 7 Number of timesteps to find 3 targets in a 30 m x 30 m arena. The results for the random search are in green and the results for the belief space search are in orange. The BF of the Bayesian paired samples t-test for each setup are respectively 66550.284, for 15 robots, 914.282 for 20 robots, and 25.268 for 25 robots. No error estimate can be given for 15 robots, however the error for 20 and 25 are respectively 4.609e-8 and 2.954e-5.

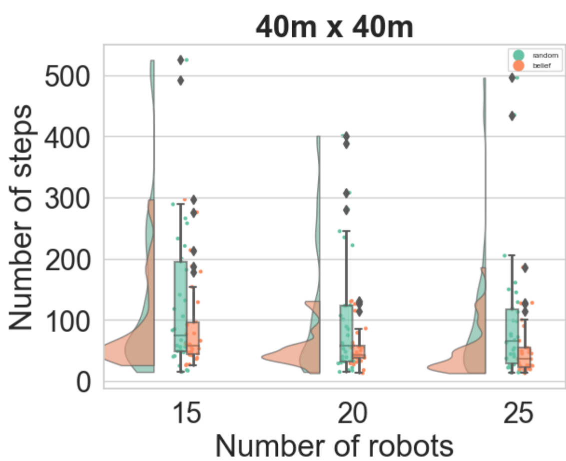

Fig. 8 Number of timesteps to find 3 targets in a 40 m x 40 m arena. The results for the random search are in green and the results for the belief space search are in orange. The BF and the error of the Bayesian paired samples t-test for each setup are respectively 210.765 and 1.358e-6, for 15 robots, 203.165 and 3.369e-7 for 20 robots, 3577.307 for 25 robots. No error estimate can be given for 25 robots.

the total time spent by the root node in the auctioneer state. Figures 9, 10, and 11 summarize the results obtained from that metric. It shows in most cases a slightly higher formation time for the belief based search, with a higher standard deviation. Those results hints that the relay formation might take more time for the belief based search, refuting our hypothesis. To get a clearer answer, we once again did a Bayesian paired samples t-test on the statistical results considering the null hypothesis. The results show that 89% of the data reject the null hypothesis with the Bayesian Factor lower than 1 (BF min = 2.539e-6, BF max = 3.862).

We can also notice a constant, but slight increase of the formation time when the arena gets bigger. That fact is expected as the searching area increases. In fact, we have less chances of finding a drone in communication range of the root since they will now search further from the root. Another interesting result is that the formation time do not change much when we compare for the same arena size, different number of drones. Here, the number of agents do not impact the results as the auction time is based on whether a winner is found or not. Thus, more agents do not change the ability for the auctioneer to find a winner, assuming they are in range.

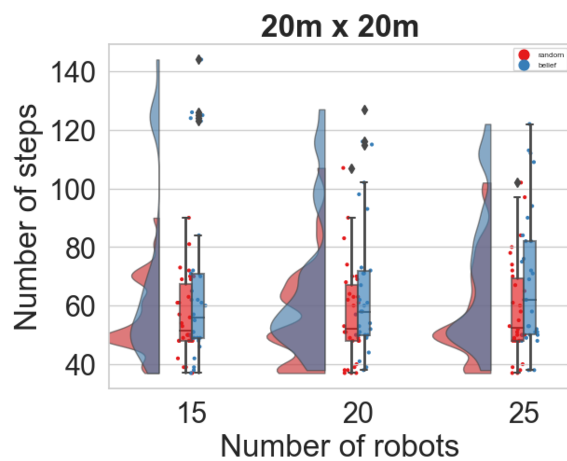

Fig. 9 Number of timesteps needed per experiment to obtain the whole relay structure in a 20 m x 20 m arena. The results for the random search are in red and the results for the belief space search are in blue. The BF and the error of the Bayesian paired samples t-test for each setup are respectively 0.115 and 2.281e-6, for 15 robots, 8.070e-4 and 1.277e-6 for 20 robots, 2.539e-6 and 2.511e-8 for 25 robots.

To explain the relay formation time difference between the two search methods, we made the hypothesis that the network usage is at the root of this observation. In fact, because of the dynamic

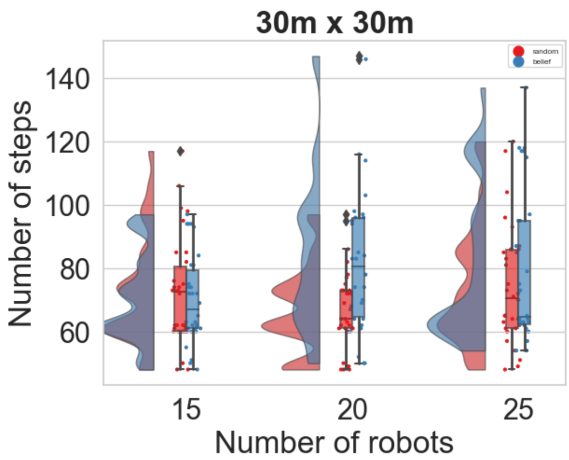

Fig. 10 Number of timesteps needed per experiment to obtain the whole relay structure in a 30 m x 30 m arena. The results for the random search are in red and the results for the belief space search are in blue. The BF and the error of the Bayesian paired samples t-test for each setup are respectively 0.264 and 3.022e-6, for 15 robots, 3.327e-6 and 2.817e-8 for 20 robots, 0.022 and 1.896e-4 for 25 robots.

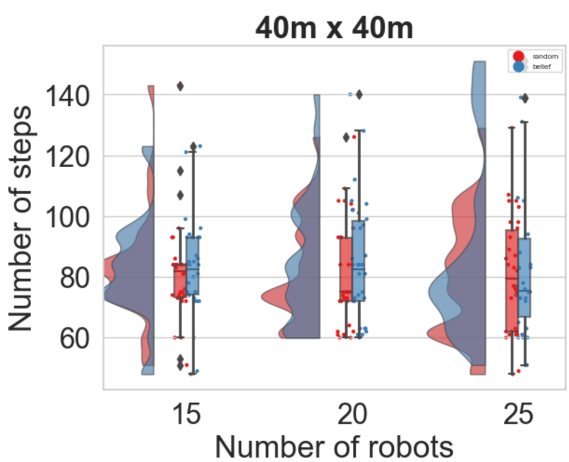

Fig. 11 Number of timesteps needed per experiment to obtain the whole relay structure in a 40 m x 40 m arena. The results for the random search are in red and the results for the belief space search are in blue. The BF and the error of the Bayesian paired samples t-test for each setup are respectively 3.862 and 0.001, for 15 robots, 0.017 and 1.133e-4 for 20 robots, 0.469 and 3.806e-6 for 25 robots.

update of the belief map, the network is heavily used in the belief based search. The update messages could delay the handling of the auction-related ones and therefore extend the formation time. To confirm this hypothesis, we compiled the bandwidth usage for the belief based search. For the sake of brevity, we present in figure 12 the result of a randomly selected experiment's configuration. We can see on the graph that for the belief-based search, the bandwidth is constantly increasing and we observe an homogeneous network usage by all the agents. In some rare occasions, when a drone would be out of range, we can observe a decrease of its network usage which increases drastically when he is back in range. Those observations confirmed our hypothesis by showing a fairly high network usage in the belief search.

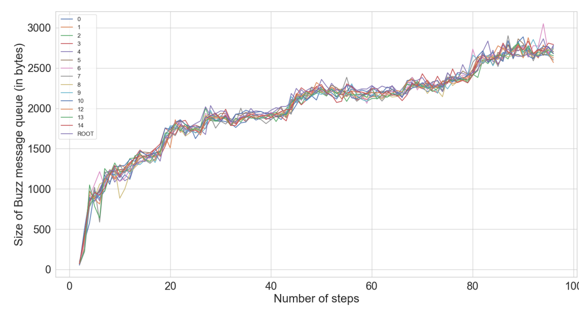

Fig. 12 Bandwidth usage for 15 drones in a 20 x 20 arena (single experiment)

It should also be noted that the number of search steps (set to 20 for our tests) could impacts the time needed to have the relay chain, showing the importance for parameters tuning. For example, if the target is found by a drone at step 1, the first relays will not come to the meeting point before step 20 (unless they also find the target or are in *communication range* of the root). That fact can make the system stabilization unnecessarily long.

5 Experiments

The real-world experiments were performed in an outdoor football field with three DJI M300 quadcopters. The search area considered for the tests is a 36 x 36 meters field. The experimental platform was mainly presented in section 3.3. The decentralized control of the drones is achieved using the same Buzz scripts used in the simulation, with some minor changes related to the auction duration, and the M300 RTK's flight controller.

The tests were performed with both the random walk search and the distributed belief map search. Each of the experiments, were performed three times to confirm the proper functioning of the algorithm. Figure 14 confirms the feasibility of our method. After performing the previously explained search pattern, the drones adopt, as seen in the picture, a relay formation. Given that we only have one target and three drones the relay was a line from the predefined rendezvous point to the target position. Sending a message from the agent at the target position, the information can now be relayed back to the root for analysis.

To give a better idea of the searching behavior during the experiments, we present in figure 13 the path taken by the drones during two randomly selected runs (Right: random, Left: belief). That figure shows for the random search sparse lines, scattered randomly over the field, indicating a lack of pattern during the search. That lack of pattern can give fast discovery of the target under certain circumstances and a very long time in other cases. On the other hand, for a belief search, the lines are concentrated in areas with high belief.

Table 1 Real-world tests results

Experiment #	Target discovery (timesteps)	Relay formation (timesteps)
1	1159	301
2	863	324
3	1295	249

Table 1 presents, for information, the recorded metrics for the belief map search. It shows the number of timesteps needed to find the target for each of the experiments. We obtained an average 1105 timesteps with the belief search, a value that is bigger than the ones obtained in simulation. In fact, the difference with field experiments is that the number of steps required to get from one position to another is higher because of the velocity of the drones set to a relatively low value. The table also contains the necessary time to form a relay chain (as the one shown in figure 14). The mean value here is 295 timesteps.

Unlike the simulations, these experiments were only performed 3 times with a fixed target and a fixed belief map. That configuration can easily produce some biased results. Therefore, more real-world tests would be necessary to have statistically relevant data.

6 Conclusion and Future works

In this paper, we presented a novel swarm robotic system for search and rescue operations in realistic scenarios. The proposed system is fully decentralized and robust to sporadic disconnections. It was also coupled with an implementation of a distributed belief map search algorithm leveraging prior knowledge on the area to realize a faster search. Based on the results obtained in simulation, we were able to confirm that our search method performs better than a random walk. The results obtained during our simulations and the real-world experiments, confirmed the feasibility of our approach. The deployed architecture also provide a modular, easily portable and scalable system, that could be used in other swarm deployments. In future work, we will leverage the distributed belief map in the search algorithm to perform dynamic updates on the target location belief. Indeed, we could model the target motion (e.g. due to the flow of a river) to update the prior over time.

Declarations

Funding. This work was supported by the Fonds de recherche du Quebec – Nature et technologies (FRQNT) under grant 296737 and by the National Research Council Canada (NRC).

References

Alotaibi ET, Alqefari SS, Koubaa A (2019) Lsar: Multi-uav collaboration for search and rescue missions. IEEE Access 7:55,817–55,832

Andries M, Charpillet F (2013) Multi-robot exploration of unknown environments with identification of exploration completion and post-exploration rendezvous using ant algorithms. In: 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, pp 5571–5578

Andries M, Charpillet F (2015) Multi-robot taboo-list exploration of unknown structured environments. In: 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, pp 5195–5201

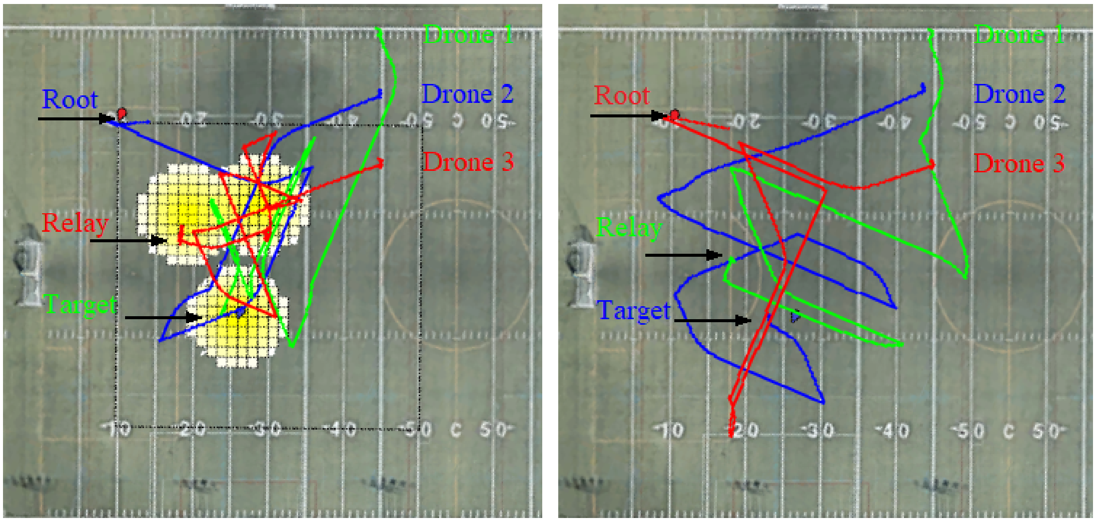

Fig. 13 GPS trajectory during tests on a football filed. Right: random, Left: belief map, the yellow cells indicate the probability of having a target

Fig. 14 Relay chain formation during real-world experiment

Apvrille L, Tanzi T, Dugelay JL (2014) Autonomous drones for assisting rescue services within the context of natural disasters. In: 2014 XXXIth URSI General Assembly and Scientific Symposium (URSI GASS), IEEE, pp 1–4

Belkadi A, Ciarletta L, Theilliol D (2016) Uavs fleet control design using distributed particle swarm optimization: A leaderless approach. In: 2016 International Conference on Unmanned Aircraft Systems (ICUAS), IEEE, pp 364–371

Brambilla M, Ferrante E, Birattari M, et al (2013) Swarm robotics: a review from the swarm engineering perspective. Swarm Intelligence 7(1):1–41

Çeltek SA, Durdu A, Kurnaz E (2018) Design and simulation of the hierarchical tree topology based wireless drone networks. In: 2018 International Conference on Artificial Intelligence and Data Processing (IDAP), IEEE, pp 1–5, URL http://dx.doi.org/10.1109/IDAP.2018.8620755

Dimidov C, Oriolo G, Trianni V (2016) Random walks in swarm robotics: an experiment with kilobots. In: International Conference on Swarm Intelligence, Springer, pp 185–196

Dorigo M, Floreano D, Gambardella LM, et al (2013) Swarmanoid: a novel concept for the study of heterogeneous robotic swarms. IEEE Robotics & Automation Magazine 20(4):60–71

Gerkey BP, Mataric MJ (2002) Sold!: Auction methods for multirobot coordination. IEEE transactions on robotics and automation 18(5):758–768

Hentati AI, Fourati LC (2020) Comprehensive survey of uavs communication networks. Computer Standards & Interfaces 72:103,451

Hourani H, Hauck E, Jeschke S (2013) Serendipity rendezvous as a mitigation of exploration's interruptibility for a team of robots. In: 2013 IEEE International Conference on Robotics and Automation, IEEE, pp 2984–2991

JASP Team (2021) JASP (Version)[Computer software]. URL https://jasp-stats.org/

Khan A, Yanmaz E, Rinner B (2014) Information merging in multi-uav cooperative search. In: 2014 IEEE international conference on robotics and automation (ICRA), IEEE, pp 3122–3129

Kim J, Ladosz P, Oh H (2020) Optimal communication relay positioning in mobile multi-node networks. Robotics and Autonomous Systems 129:103,517. https://doi.org/https://doi.org/10.1016/j.robot.2020.103517, URL https://www.sciencedirect.com/science/article/pii/S0921889019309145

Kiran K, Kaushik N, Sharath S, et al (2018) Experimental evaluation of batman and batman-adv routing protocols in a mobile testbed. In: TENCON 2018-2018 IEEE Region 10 Conference, IEEE, pp 1538–1543

Kobayashi F, Sakai S, Kojima F (2002) Sharing of exploring information using belief measure for multi robot exploration. In: 2002 IEEE World Congress on Computational Intelligence. 2002 IEEE International Conference on Fuzzy Systems. FUZZ-IEEE'02. Proceedings (Cat. No. 02CH37291), IEEE, pp 1544–1549

Kobayashi F, Sakai S, Kojima F (2003) Determination of exploration target based on belief measure in multi-robot exploration. In: Proceedings 2003 IEEE International Symposium on Computational Intelligence in Robotics and Automation. Computational Intelligence in Robotics and Automation for the New Millennium (Cat. No. 03EX694), IEEE, pp 1545–1550

Li J (2019) Throughput-aware flying communication relay network for disaster area search and rescue. In: Proceedings of the 2019 8th International Conference on Networks, Communication and Computing, pp 138–141, URL http://dx.doi.org/10.1145/3375998.3376038

Majcherczyk N, Jayabalan A, Beltrame G, et al (2018) Decentralized connectivity-preserving deployment of large-scale robot swarms. In: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, pp 4295–4302

Nickerson JV (2004) Robots and humans reconvening. In: 2004 IEEE International Conference on Systems, Man and Cybernetics (IEEE Cat. No. 04CH37583), IEEE, pp 2803–2808

Pinciroli C, Beltrame G (2016) Buzz: An extensible programming language for heterogeneous swarm robotics. In: 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, pp 3794–3800

Pinciroli C, Trianni V, O'Grady R, et al (2012) Argos: a modular, parallel, multi-engine simulator for multi-robot systems. Swarm intelligence 6(4):271–295

Pinciroli C, Lee-Brown A, Beltrame G (2016) A tuple space for data sharing in robot swarms. In: Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies (formerly BIONETICS), pp 287–294

Quigley M, Conley K, Gerkey B, et al (2009) Ros: an open-source robot operating system. In: ICRA workshop on open source software, Kobe, Japan, p 5

Ruetten L, Regis PA, Feil-Seifer D, et al (2020) Area-optimized uav swarm network for search and rescue operations. In: 2020 10th Annual Computing and Communication Workshop and Conference (CCWC), IEEE, pp 0613–0618, URL http://dx.doi.org/10.1109/CCWC47524.2020.9031197

St-Onge D, Varadharajan VS, Li G, et al (2017) Ros and buzz: consensus-based behaviors for heterogeneous teams. arXiv preprint arXiv:171008843

St-Onge D, Kaufmann M, Panerati J, et al (2019) Planetary exploration with robot teams: Implementing higher autonomy with swarm intelligence. IEEE Robotics & Automation Magazine 27(2):159–168. URL http://dx.doi.org/10.1109/MRA.2019.2940413

Tarapore D, Groß R, Zauner KP (2020) Sparse robot swarms: moving swarms to real-world applications. Frontiers in Robotics and AI 7:83

Varadharajan VS, St-Onge D, Adams B, et al (2020) Swarm relays: Distributed self-healing ground-and-air connectivity chains. IEEE Robotics and Automation Letters 5(4):5347–5354

Vielfaure D, Arseneault S, Lajoie PY, et al (2021) Dora: Distributed online risk-aware explorer. arXiv preprint arXiv:210914551

Wubben J, Aznar P, Fabra F, et al (2020) Toward secure, efficient, and seamless reconfiguration of uav swarm formations. In: 2020 IEEE/ACM 24th International Symposium on Distributed Simulation and Real Time Applications (DS-RT), IEEE, pp 1–7

Yamaguchi SP, Karolonek F, Emaru T, et al (2017) Autonomous position control of multi-unmanned aerial vehicle network designed for long range wireless data transmission. In: 2017 IEEE/SICE International Symposium on System Integration (SII), IEEE, pp 127–132, URL http://dx.doi.org/10.1109/SII.2017.8279200

Zhang HG, Jin GY, Qu YX, et al (2021) Servo relays as distributed controllable-mobility network to maintain long-term stable links for mobile robot swarms. Ad Hoc Networks 117:102,497. https://doi.org/https://doi.org/10.1016/j.adhoc.2021.102497, URL https://www.sciencedirect.com/science/article/pii/S1570870521000597