Journal of Intelligent & Robotic Systems (2024) 110:90
https://doi.org/10.1007/s10846-024-02097-0

REGULAR PAPER

Restoring Connectivity in Robotic Swarms – *A Probabilistic Approach*

Kasra Eshaghi[1] · Naeimeh Najafizadeh Sari[1] · Cameron Haigh[1] · Darie Roman[1] · Goldie Nejat[1] · Beno Benhabib[1]

Received: 16 October 2023 / Accepted: 25 March 2024 / Published online: 13 June 2024

Abstract

Connectivity is an integral trait for swarm robotic systems to enable effective collaboration between the robots in the swarm. However, connectivity can be lost due to events that could not have been a priori accounted for. This paper presents a novel probabilistic connectivity-restoration strategy for swarms with limited communication capabilities. Namely, it is assumed that the swarm comprises a group of *follower* robots whose global connectivity to a base can only be achieved via a localized *leader* robot. In this context, the proposed strategy incrementally restores swarm connectivity by searching for the lost robots in regions-of-interest (RoIs) determined using probability theory. Once detected, newly found robots are either recruited to help the leader in the restoration process, or directly guided to their respective destinations through accurate localization and corrective motion commands. The proposed swarm-connectivity strategy, thus, comprises the following three stages: (*i*) identifying a discrete set of optimal RoIs, (*ii*) visitation of these RoIs, by the leader robot, via an optimal inter-region search path, and (*iii*) searching for lost robots within the individual RoIs via an optimal intra-region search path. The strategy is novel in its use of a probabilistic approach to guide the leader robot's search as well as the potential recruitment of detected lost robots to help in the restoration process. The effectiveness of the proposed probabilistic swarm connectivity-restoration strategy is represented, herein, through a detailed simulated experiment. The significant efficiency of the strategy is also illustrated numerically via a comparison to a competing random-walk based method.

Keywords Swarm robotics · Connectivity restoration · Motion planning · Probabilistic modelling

Classification Codes 93C85 · 93A16

1 Introduction

Swarm robotic systems (SRSs) represent teams of large number of robots that collaborate to accomplish complex tasks [1–4] such as environmental monitoring [5], collective perception [6], and exploration [7]. Collaboration is, typically, achieved through exchange of information amongst the member robots as well as between the robots and possible external infrastructure via wireless communication devices and/or onboard sensors [8–15]. In order to achieve effective communication, however, swarm members must maintain a desired degree of *connectivity*, which specifies for each member what other teammates and/or external infrastructure it should be able to communicate with (*e.g.*, [16]). Real-time connectivity maintenance, however, may be a challenging problem due to limited sensing/ communication range, constrained line-of-sight, interference, etc.

Connectivity must, thus, be considered while planning the motion of the swarm members, for example, via *online* relocation of the member robots when they get close to their communication limits [17–24], or through *offline* constrained trajectory planning [25–27]. While such methods may successfully maintain connectivity in 'controlled' environments, they may fail when unexpected/unplanned-for events occur; including, loss of member robots due to hardware failure, obstacles in the environment whose positions were a priori unknown, or errors in the execution of (robot) motion commands. Such events would result in the swarm being disconnected, necessitating the implementation of a (connectivity) *restoration* process for the (disconnected) 'lost' robots.

✉ Beno Benhabib
benhabib@mie.utoronto.ca

[1] Department of Mechanical and Industrial Engineering, University of Toronto, 5 King's College Rd, Toronto, ON M5S 3G8, Canada

Connectivity-restoration is, typically, a two-stage process that involves (1) estimating the state of all swarm robots and, then, (2) planning their motion to new locations to re-establish connectivity. The 'state' of the robots could include their own positions and/or a model of their surrounding environment. Though, as will be discussed below, some existing methodologies ignore the state-estimation step, and directly plan the robots' motion independently of an estimate of the swarm's state.

This paper presents a novel swarm-connectivity strategy that addresses the connectivity-restoration problem for a swarm that has moved from an initial configuration/formation to another configuration some tangible distance away. Specifically, we consider a swarm comprising a *leader robot* with accurate localization and motion capabilities, achieved through (*i*) GPS technology, (*ii*) complex onboard sensors (*e.g*., cameras) coupled with a map of the operating environment, or (*iii*) an environment that includes active markers (*e.g*., beacons) that can localize the leader robot. The leader robot guides several *follower robots* that do not have accurate localization and are subject to cumulative random errors in the execution of their motion. These errors may directly lead to disconnectivity in the swarm when arriving at its desired destination. Furthermore, they present a challenge as (1) the positions of the disconnected follower robots must be estimated (with uncertainty), and (2) the accumulation of random errors in the execution of their motion commands must be considered in the restoration process.

A review of the pertinent literature and the contribution of this work are detailed in Section 2. The connectivity-restoration problem considered herein is formulated in Section 3, and the proposed solution methodology is detailed in Section 4, respectively. Section 5 presents an in-depth illustrative example, and compares the proposed strategy to a competing random-walk based method. The paper is concluded in Section 6.

2 Literature Review

Existing works on connectivity restoration are first discussed in detail in Section 2.1 below. In some pertinence to the objective at hand, a review of connectivity restoration in the context of wireless sensor networks (WSN) is provided next in Section 2.2, and a review of the multi-target search problem, whose formulation can be applied to the connectivity-restoration problem addressed in this paper is provided in Section 2.3, respectively.

2.1 Swarm Connectivity Restoration

Connectivity restoration can be categorized according to the knowledge of the swarm's state used for the restoration process. They include 1) those that assume the swarm's state is accurately known, Sections 2.1.1, 2) those that estimate the state of the swarm, Sections 2.1.2, and 3) those that do not use information on the swarm's state to plan the connectivity restoration process, Section 2.1.3.

2.1.1 Swarm-State Known

Accurate knowledge of a swarm's state at any given time would simplify connectivity restoration as robots can plan their needed motion even when disconnected. In [28–30], for example, it is assumed that the trajectory of a *gateway robot* that the swarm must remain connected to is accurately known. The disconnected robots, who also know their own positions accurately, plan their motion to intercept this gateway robot (*e.g.* [31–34],). In [35], each robot calculates the centroid of the swarm based on the known position of all other robots in the swarm, and moves toward the centroid until swarm connectivity is restored. In [36], each robot is assigned a *parent* robot, and is the *child* of another robot. Once it fails, its parent and child, who know their own position and that of the failed robot accurately, move toward each other to restore the connectivity of the swarm.

While the aforementioned approaches may be suitable in scenarios where the swarm's state is accurately known, they may not be successful in restoring connectivity when the positions of the disconnected robots cannot be estimated with certainty. Namely, when the positions of the disconnected robots are not known with certainty, such methods may plan motions that do not restore connectivity. These approaches also do not consider cumulative random errors in motion execution of the member robots.

2.1.2 Swarm-State Estimated

Swarm-state estimation in the context of connectivity restoration was addressed in [37] and [38]. In [37], the authors consider a scenario where the swarm loses connectivity due to a priori unknown obstacles that are encountered during task execution. It is assumed that all robots have accurate knowledge of their own positions and those of their peers. They use this information to estimate the environment of their peers and plan their motion to intercept them and restore connectivity.

Loss of swarm connectivity due to intermittent communication failure was considered in [38]. In this work, a disconnected robot estimates the direction of the rest of the swarm, based on the swarm's learned behavioral model, and plans its own motion to intercept the swarm. This work assumes that member robots have accurate knowledge of their own position.

Both above approaches, [37] and [38], assume the member robots have accurate knowledge of their positions, which would also allow them to compensate for cumulative random errors in motion execution. Furthermore, they do not consider potential uncertainties in the estimated state of the swarm.

2.1.3 Swarm-State not Used

Methods that plan robot motion without relying on an estimate of the swarm's latest state have also been suggested, including rendezvous-based and behavior-based methods. Rendezvous-based methods select a *meetup point* that the robots should move to when connectivity is lost [39–41]. This meetup point can be selected before the swarm begins executing its task, either arbitrarily [39] or as the point of swarm deployment [40]. It may also be selected in an online manner based on a shared meetup point selection policy. This approach was used in [41], where the disconnected robots selected a meetup point as the most distinct landmark in their surrounding environment, where it is assumed that each landmark has an inherent distinctiveness measure that can be sensed by the robots, and that all robots have seen this landmark.

Behavior-based methods design *ad-hoc* control policies for the robots to use for restoring connectivity [42–44]. These include backtracking to the last position where the swarm was connected [42, 43], moving to areas in the environment where the robot may have a larger communication range (*e.g.,* open spaces/higher altitudes) [42], and random walks [44].

While above works do not use an estimate of the swarm's latest state to restore connectivity, they do require that the robots accurately know their own positions. This allows them to compensate for cumulative random errors in motion execution and to converge to a state that restores the connectivity of the swarm. When this condition is not met – namely, when robots have uncertainty in their estimated positions and are subject to cumulative random errors in the execution of their motion commands – the robots would not be able to accurately move to their selected meetup point or to execute the desired behavior. This may result in the robots moving to destinations that do not restore connectivity.

2.2 Wireless-Sensor Networks

Connectivity restoration has also been addressed in the wireless-sensor networks (WSNs) literature [45–52]. In such networks, it is assumed that the nodes know their own positions accurately and can share this information with their neighbors through wireless communication channels. Connectivity, however, may be lost due to node failure, resulting in the partitioning of the network into multiple sub-networks.

Solutions to connectivity restoration in WSNs by deploying relay nodes have been discussed in [45, 46], or moving the partitioned sub-networks toward each other until connectivity is restored in [47], respectively. The use of *caretaker* nodes has also been suggested, whose role is to move to the position of the failed node for restoring the connectivity of the network [48–52]. Such works, typically, address the problems of detecting node failure, determining whether the failed node was a critical node (*i.e.,* whose failure results in the partitioning of the network into multiple disconnected sub-networks), and selecting a caretaker node in an online manner.

Approaches discussed above for restoring connectivity in WSNs, although addressing a similar problem to swarm disconnectivity, could not be easily applied to the problem addressed in this paper as they assume the positions of all nodes in the network are accurately known, and that the nodes can move to their planned destinations accurately.

2.3 Multi-Target Search

The multi-target search literature deals with the problem of using a single or multiple autonomous agents, who have accurate localization capabilities, to search for multiple static or dynamic targets, whose positions are not accurately known [53–58]. Such formulations have not been, commonly, applied to swarm connectivity restoration. However, its similarities to the connectivity restoration problem suggest that a review of the literature in this field is appropriate.

There are generally two approaches to the multi-target search problem. The first approach is to obtain an estimate of the potential locations of all targets, and to merge these estimates into an overall map that is used to guide the trajectory of the agents. This approach is taken in [53], where the merged map of the potential locations of the targets is represented through a probability density function. A grid is then superimposed onto this function, where the value of each cell in this grid corresponds to the probability of detecting any of the targets. The authors propose two strategies to allocating the cells to the agents: to maximize the expected probability of finding a target and to maximize the minimum probability of target detection. A similar approach was taken in [54], where multi-target detection in the context of post-earthquake search and rescue in an urban environment is addressed. A grid, superimposed onto a probability density function, is used as the map of the environment, and this map is explored by the (single) agent through a greedy search.

A more complex approach proposed in the literature is to use an estimate of the locations of individual targets to plan the search [55–58]. The work in [55] addresses the problem of retrieving drifting sensors that are used for ocean monitoring. They model the estimated position of a drifting sensor as a circular area, centered at its expected position, determined based on wind and current patterns of their operating environment. The developed search strategy includes a search pattern and a search schedule. The search pattern selects the trajectory that the (single) agent takes to explore the area of the sensor, for which a spiral pattern is suggested. The search schedule, in turn, redirects the agent from searching for one sensor to another based on a regular time interval. This method was extended in [56] to model the estimated positions of the sensors using probability density functions, and to redirect the agents to the next nearest sensor once the cumulative probability of detection of the current sensor distribution exceeds a threshold.

The use of multiple agents for multi-target search was also addressed in [57]. The authors allocate the agents to the targets to minimize a cost function that is dependent on the distance between the agents and the targets, and the radius of the area that represents the potential location of each target. The radius of the area is obtained through an empirically derived function. This objective allows them to consider the additional effort required to search for targets whose positions are more uncertain than others. The problem of marine trash collection was addressed in [58]. In this work, it is proposed to cluster targets that are estimated to be close to each other, and to allocate agents to clusters rather than individual targets. Agent allocation is completed through a cost function that depends on the distance between the agent and the clusters, the size of the cluster, and the total number of targets in the cluster. The size of the cluster is obtained through an empirically derived function.

The difference between the multi-target search problem and swarm connectivity restoration is the onboard capabilities of the retrieved targets. Namely, in the multi-target search problem, the targets do not have onboard sensors, in contrast to the retrieved disconnected robots in a swarm that are equipped with sensors that allow them to detect each other. In a swarm, thus, the retrieved disconnected robots can be used to help restore connectivity to the remaining robots. This difference calls for the development of a restoration strategy that leverages the sensing capabilities of the retrieved agents.

2.4 Challenges and Contributions

This paper addresses the problem of connectivity restoration for swarms comprising robots that do not have accurate localization capabilities and are subject to random errors in the execution of their motion. As noted in the Introduction, a connectivity-restoration strategy for such swarms must be developed to consider these two challenges.

The connectivity-restoration literature, typically, does not consider the abovementioned challenges. Namely, as noted in Sub-Sections 2.1.1–2.1.3, the pertinent literature [28–30, 35–44] does not consider uncertainty in the estimated positions of the swarm, and/or requires the robots to execute the motion commands without cumulative random errors. Similarly, approaches to restoring connectivity in WSNs are limited for the same reason [45–52]. Thus, these cannot be adopted for the problem addressed in this paper.

Approaches developed for the multi-target search problem [53–58] can be applied to the problem at hand. Namely, such approaches consider the uncertainty in the estimated position of the targets to be retrieved by representing their positions through probability density functions [53, 54, 56] or through regions whose areas are empirically derived [55, 57, 58]. They, then, retrieve the targets using agents that have accurate onboard localization and motion control capabilities.

The connectivity-restoration strategy proposed herein follows a similar principle. Namely, once disconnected, the proposed strategy uses the leader robot who has accurate localization capabilities to retrieve the disconnected follower robots that remain stationary. The uncertain positions of the disconnected follower robots are represented through a probabilistic approach. As first introduced in our previous work on wilderness search and rescue [59–63], such approaches have been shown to allow for planning optimal paths for (lost) target detection. The proposed connectivity-restoration strategy is, thus, novel in that it makes use of a probability-theory based approach to achieve efficient swarm connectivity restoration. By using a probabilistic approach, the proposed strategy can (*i*) bound the region to be explored by the leader robot and provide a probabilistic guarantee of restoring connectivity, and (*ii*) allow the leader robot to prioritize areas that have higher probabilities of detecting disconnected members.

The proposed strategy also leverages the sensing capabilities of the retrieved follower robots by using them to detect the remaining robots that remain disconnected. Namely, once a follower robot is detected by the leader, it may be used to create a search team that collectively restores connectivity to the remaining disconnected robots. The use of retrieved agents for expediting the restoration process has not been explored in the multi-target search literature.

3 Problem Definition

This paper addresses the connectivity-restoration problem for a robotic swarm that moves from an initial configuration to a (next) *desired configuration*, all defined with respect to a global reference frame, ${}^{G}F$, Fig. 1. It is assumed that intermittent swarm localization may be achieved via a *leader* robot (indicated by a black outline in Fig. 1) that has full localization capabilities with respect to ${}^{G}F$, achieved through (*i*) GPS technology, (*ii*) complex onboard sensors (*e.g.*, cameras) coupled with a map of the operating environment, or (*iii*) an environment that includes active markers (*e.g.*, beacons). However, due to time-consuming real-time communication requirements,

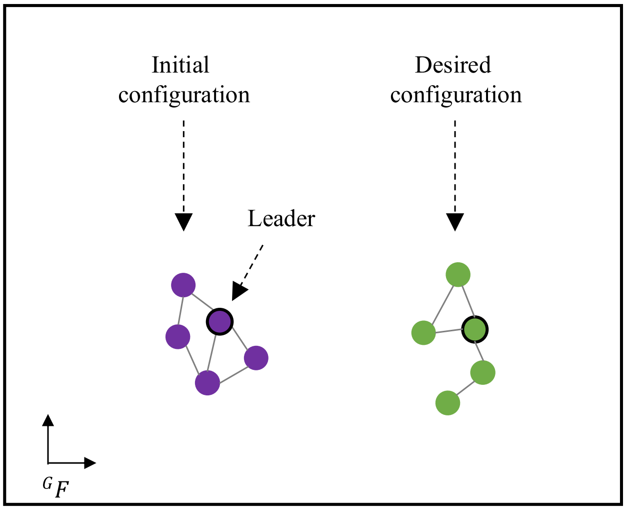


Fig. 1 Point-to-point motion of the swarm

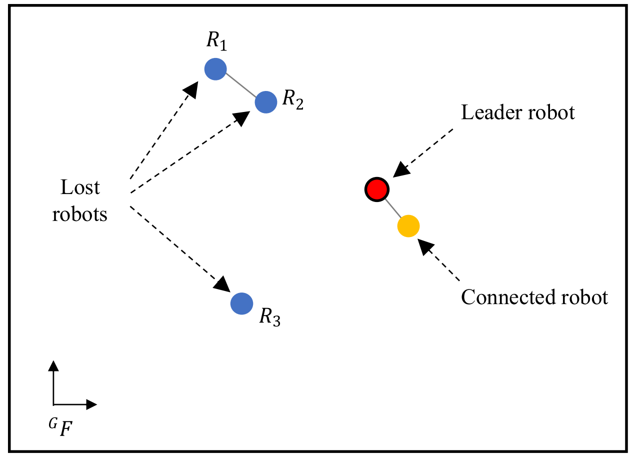


Fig. 2 A disconnected swarm state

especially, for large-sized swarms, member (*follower*) robots move to their next desired positions individually, without attempting to maintain communication with the leader robot during their travel. Namely, only upon arrival of all, the leader robot is tasked with localizing the swarm through inter-robot communication. Such localization, though, would require complete connectivity. Namely, upon reaching the desired configuration, all follower robots must be able to communicate with the leader robot, directly or indirectly. However, it is often likely that since follower robots are subject to cumulative motion errors, potential disconnectivities would need to be dealt with upon arrival at the desired swarm configuration. In such a scenario, a connectivity-restoration process must be invoked.

Once the swarm enters a disconnected state, the follower robots are designated as either *lost* or *connected*, Fig. 2. Lost robots are those that are not connected to the leader robot, directly or indirectly. It is assumed that at this disconnected state, the leader robot would explore its surrounding environment to detect/find the lost robots, who in turn would remain stationary until being detected by the leader robot.

It is expected that the leader robot would restore connectivity by selectively searching for the lost robots. Although initially only the leader robot is designated as the searcher for the lost robots, once a lost robot is detected during the search, it may, in turn, also be recruited to help the leader robot in its search for the remaining lost robots.

The connectivity restoration problem at hand, thus, comprises four main subproblems: (*i*) examining the search space for identifying a discrete set of regions-of-interest (RoIs), as will be discussed below in Sub-Section 3.1, (*ii*) sequencing of the visitations of these RoIs by the leader robot (*i.e.*, inter-region search path planning), Sub-Section 3.2, (*iii*) searching for lost robots within the individual RoIs (*i.e.*, intra-region search path planning), Sub-Section 3.3, and (*iv*) combination of the solutions to Sub-problems (*i*) to (*iii*) into an effective on-line search algorithm for efficient swarm-connectivity establishment, Sub-Section 3.4.

3.1 Examining the Search Space – Identifying Regions-of-Interest

The overall objective of the proposed strategy for swarm connectivity, addressed herein, is to address the problem in an effective and time-efficient manner. In this regard, fast connectivity restoration could be achieved by minimizing the search area. Such a goal, in turn, could be realized by designating a limited number of regions-of-interest (RoIs) to narrow the search for the lost robots.

One can use probability theory to identify RoIs by examining the motions of the robots in the swarm. For example, a RoI for an individual lost robot can be defined using a probabilistic model of the position of that robot. This model can be obtained based on the robot's desired position in the swarm configuration where connectivity was lost, the motion commands executed in getting to this position, its motion model, and the map of the environment that details the positions of surrounding obstacles. It is assumed, herein, that such a map is readily available.

In the above context, one may even consider the grouping of lost robots in order to minimize search redundancy. Namely, if the RoIs of multiple lost robots intersect significantly, then, their corresponding RoIs could be merged. The combined RoI would be a contiguous region where any of the corresponding lost robots can be detected. Such a process would reduce the size of the set of RoIs.

Let $\boldsymbol{G} = \left\{G_i\right\}_{i=1}^{n_G}$ represent the grouping of all the lost robots into n_G groups (*i.e.*, RoIs), where G_i represents the

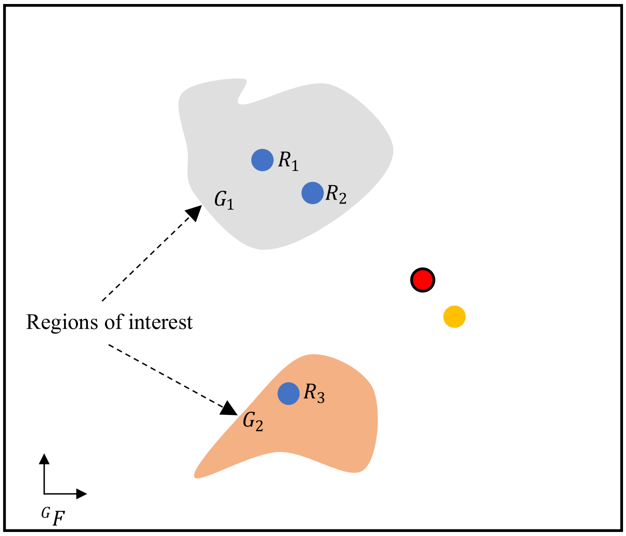


Fig. 3 Regions-of-interest (RoIs)

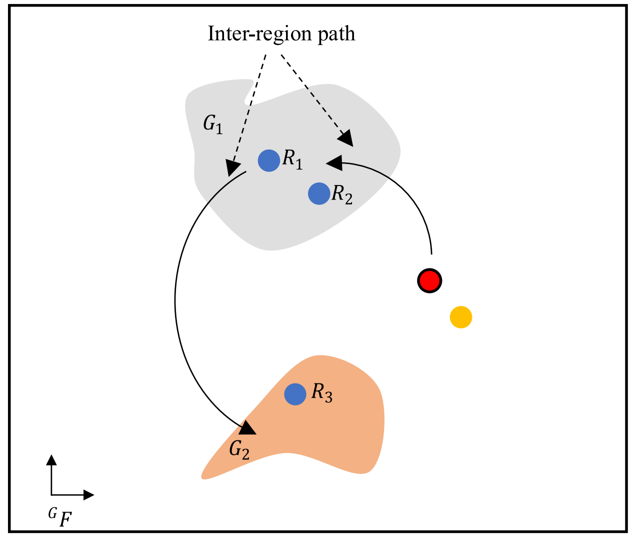


Fig. 4 Inter-region path

lost robots in Group i. Figure 3 shows the RoIs for an example grouping solution for the disconnected swarm considered in Fig. 2, where $G_1 = \{R_1, R_2\}$ and $G_2 = \{R_3\}$.

Lost-robot grouping (*i.e.*, identification and merging of the RoIs) must determine the optimal combination of the lost robots, $\boldsymbol{G}^*$ (*i.e.*, optimal set of RoIs), to minimize the total search area, $A_T(\boldsymbol{G})$, that must be explored to restore the connectivity of the swarm:

$$\min A_T(\boldsymbol{G}). \tag{1}$$

3.2 Inter-Region Search Path Planning – Solving the Travelling-Salesperson Problem

The inter-region path planning stage of the proposed strategy would deal with the problem of selecting the order in which the RoIs should be visited by the leader robot. In this regard, let $\boldsymbol{P} = \{P_i\}_{i=1}^{n_p}$ represent the inter-region path, where P_i is the i^{th} RoI that is visited by the leader robot, and n_p is the number of RoI that must be visited. Figure 4 illustrates an example inter-region path/string for the example at hand, where the leader robot visits G_1 first and, then, G_2, *i.e.*, $P_1 = G_1$, $P_2 = G_2$, and $\boldsymbol{P} = \{G_1, G_2\}$.

The inter-region path-planning problem is an instance of the classical travelling-salesperson problem. In the typical version of the traditional problem, the string of cities to be visited is selected to optimize the total distance travelled (*i.e.*, travel time). In our problem herein, the (leader) search robot is the 'salesperson', the RoIs represent the 'cities' to be visited, and the time that must be spent at each city is equivalent to the *search effort* that must be expended to explore the RoI at hand.

It is, thus, proposed herein to select the optimal inter-region path, $\boldsymbol{P}^*$, that minimizes an overall objective function that is a weighted combination of the search effort described above, and the distance that the leader robot would need to travel to move from one RoI to another along its inter-region path:

$$\min J(\boldsymbol{P}) = \alpha \frac{D(\boldsymbol{P})}{D_{max}} + \beta \frac{C(\boldsymbol{P})}{C_{max}}. \tag{2}$$

Above, $D(\boldsymbol{P})$ and $C(\boldsymbol{P})$ represent *total travel distance* and the *total search effort*, respectively. Both measures are normalized with respect to their maximum values, D_{max} and C_{max}, and are weighted according through α and β, where $\alpha + \beta = 1$.

3.3 Intra-Region Search Path Planning – Using Probability Theory for Finding Lost Robots

The intra-region path-planning stage of the proposed strategy must address the determination of an efficient search path to explore a RoI at hand. An optimal search path would need to be planned for the leader robot. Since a RoI, as defined in Section 3.1, is formulated in terms of a probabilistic model of the locations of the lost robots, the intra-region search can also be formulated via a probabilistic method.

A successful search would be one that results in connectivity being restored between the leader robot and the lost robots in the RoI as fast as possible. As such, the probability of successfully searching a RoI associated with lost robot group G_j, POS_{G_j}, can be defined as the probability that connectivity is successfully restored to all robots in the group:

$$POS_{G_j} = \prod_{R_i \in G_j} POR_{R_i}, \tag{3}$$

where POR_{R_i} is the probability of connectivity being restored with lost Robot R_i, within the group/RoI being searched for, G_j.

POR_{R_i} can be defined as the product between the probability of that lost robot being in the detection area of the leader robot, POA_{R_i}, and the probability of detecting that lost robot, POD_{R_i}:

$$POR_{R_i} = POA_{R_i} \times POD_{R_i}. \tag{4}$$

For any given leader robot detection area, A_D, and lost Robot position, ${}^G\boldsymbol{x}_{R_i}$, the POA_{R_i} can be defined as the probability of the position being inside the detection area:

$$POA_{R_i} = P\left({}^G\boldsymbol{x}_{R_i} \in A_D\right). \tag{5}$$

Above, POD_{R_i} can also be defined in terms of ${}^G\boldsymbol{x}_{R_i}$ and A_D as well as the variable D_{R_i} which denotes if connectivity restoration with lost Robot R_i was successful:

$$POD_{R_i} = P\left(D_{R_i} = success | {}^G\boldsymbol{x}_{R_i} \in A_D\right). \tag{6}$$

This would, in turn, yield:

$$POR_{R_i} = P\left(D_{R_i} = success, {}^G\boldsymbol{x}_{R_i} \in A\right). \tag{7}$$

Since the overall objective is to minimize the time spent to restore connectivity, the search should be efficient in terms of maximizing the probability of success. This can be achieved by following a search path where the time spent searching an area is proportional to the density of the probabilistic model of the position of the lost robot in the group at hand. Namely, locations that have a higher likelihood of containing the lost robots should be searched proportionally longer periods of time. Percentiles can be used to guide such a search, where searching an increasing percentile of the lost robot distribution will ensure that higher density areas are searched proportionally more and serve to distribute the locations where the search is being conducted. As such, the generated intra- region path should follow:

$$p(t) \propto E(t), \tag{8}$$

where $p(t)$ is the percentile of the lost robot locations being searched at time t and $E(t)$ is a measure of the cumulative time and resources expended up to time t.

Figure 5 illustrates an example intra-region path for the example at hand.

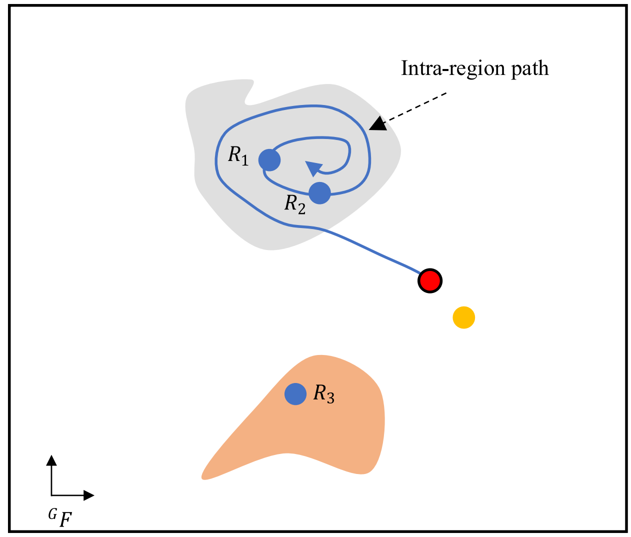


Fig. 5 An intra-region search path

3.4 Search Execution

It is expected that execution of the planned overall search path would lead to the detection of lost robot(s) with some probabilistic certainty. In an ideal scenario, the leader robot would detect the lost robots and may recruit them, temporarily, to help in searching for the remaining lost robots or guide them to their positions in the desired configuration through corrective motion commands. In practice, however, the executed search may not proceed as planned, in which case the RoIs and the leader robot's inter-region path may need to be replanned. At least three such potential scenarios may need to be considered:

1. A detected (lost) robot may belong to multiple RoIs.
2. The leader robot may (unintentionally) detect a lost robot that does not belong to the group corresponding to the RoI currently explored.
3. The execution of the planned intra-region path may not restore connectivity to a lost robot in the group associated with this region.

4 Proposed Methodology

The proposed swarm connectivity-restoration strategy, invoked upon detection of loss of connectivity upon arrival at a swarm configuration, assumes the existence of a leader robot whose location is always known. Herein, it is assumed that this robot would 'lead' the efforts to search for the lost disconnected/lost robots, which must remain static until they are detected.

The proposed strategy comprises three main stages: (*i*) evaluating the search space to identify a set of optimal regions-of-interest (RoIs) to be explored, (*ii*) determining the optimal path of the searcher (leader) robot for its *inter-region* travel, and, (*iii*) determining the optimal path of the searcher robot during its *intra-region* exploration for the lost robots assumed to be present in the RoI at hand.

As the search progresses, when lost robots are detected, they (*i*) may simply be provided with corrective motion commands to move expeditiously to their expected positions in the desired swarm configuration, or (*ii*) may be recruited, to join the leader robot and become part of a searcher group, to expedite the search for the remaining lost robots in the RoI at hand. Once the search of a RoI is complete, all newly detected robots and the searcher group robots would be guided to their respective positions in the desired swarm configuration, while the leader robot would move on to explore the next RoI on the inter-region path. This process, as is detailed in Sub-Sections 4.1–4.4, below, would be repeated until all RoIs have been explored, at which point the leader robot would also return to its desired position, and swarm connectivity would be re-checked. At this point, if connectivity restoration was still unsuccessful, the restoration process would be repeated.

The proposed connectivity restoration strategy is novel in its use of probability theory to plan the overall search path of the leader robot. Namely, herein, the use of iso-probability curves is uniquely proposed both for identifying the RoIs that must be explored (*i.e.,* grouping of the lost robots) and for planning the intra-region search paths used by the searcher (leader) robot to explore the identified RoIs. Such a probabilistic approach, as has been shown in our past work [59–66] for wilderness search and rescue, would yield efficient and effective swarm connectivity restoration. The proposed restoration strategy is also novel as it recruits the detected lost robots for exploring the planned RoIs. This formation of a *searcher group* would further expedite the connectivity-restoration process.

4.1 Examining the Search Space – Identifying Regions-of-Interest

As the first stage of swarm connectivity-restoration process, the task at hand is to identify the RoIs that must be explored to find the lost/disconnected robots. In our proposed strategy, RoIs are first determined for the individual lost robots using probability theory. These RoIs are subsequently examined for potential merger, by grouping the respective lost robots, if deemed beneficial. It is conjectured, herein, that grouping of robots, whose RoIs overlap significantly, may lead to tangible reduction in search redundancy and, thus, in connectivity-restoration time.

The description of the probabilistic representation of RoIs via iso-probability curves is discussed, first, in Section 4.1.1. Then the criterion for grouping lost robots based on the overlap of their corresponding RoIs is formulatedin Section 4.1.2. Lastly, the overall lost-robot grouping technique is detailed in Section 4.1.3.

4.1.1 Iso-Probability Curves for Representing Regions-of-Interest

The RoI of an individual (lost) robot can be defined using a probabilistic model of its position. Such an approach would be beneficial as it would allow the motion of the search (leader) robot to be planned by proportionally prioritizing regions with a higher probability of detecting a lost robot.

Herein, it is proposed to use *iso-probability curves* to represent the probabilistic model of the position of a lost robot. Iso-probability curves, as first introduced in [59], provide an approach to establishing bounds on a lost robot's location using estimates of its deviation from its desired position in the swarm configuration when connectivity was lost. The deviation of the robot from its desired position would be estimated based on the motion commands it executed in attempting to move to this position and its motion model subject to probabilistic uncertainties.

For a (single) lost robot, the corresponding (cumulative) iso-probability curves would be centered at its desired position. As an example, Fig. 6 illustrates the 99%, 75%, 50%, and 25% (cumulative) iso-probability curves for the lost robots (a) R_1, (b) R_2, and (c) R_3, shown previously in Fig. 2, respectively. Namely, for example, it is expected that there exists 25% chance for the lost robot to be within the area bounded by the 25% iso-probability curve, 50% chance for it to be within the area bounded by the 50% iso-probability curve, etc. The clouds of points represent the estimated probable deviations of each robot from its desired position.

In our connectivity-restoration strategy, a RoI is defined/bounded by the upper most iso-probability curve considered, denoted herein by the p_u%, where p_u is a user-defined parameter. However, in the absence of a defined statistical model, the iso-probability curves need to be generated through a two-stage process of *cloud generation* and *curve calculation*. The former stage generates a cloud of points to obtain estimates of the deviations of the lost robot from its desired position by simulating the robot's (noise-prone) motion. The latter stage, then, determines the set of iso-probability curves based on the cumulative distribution of the point cloud in all directions, as further detailed in Appendix A.

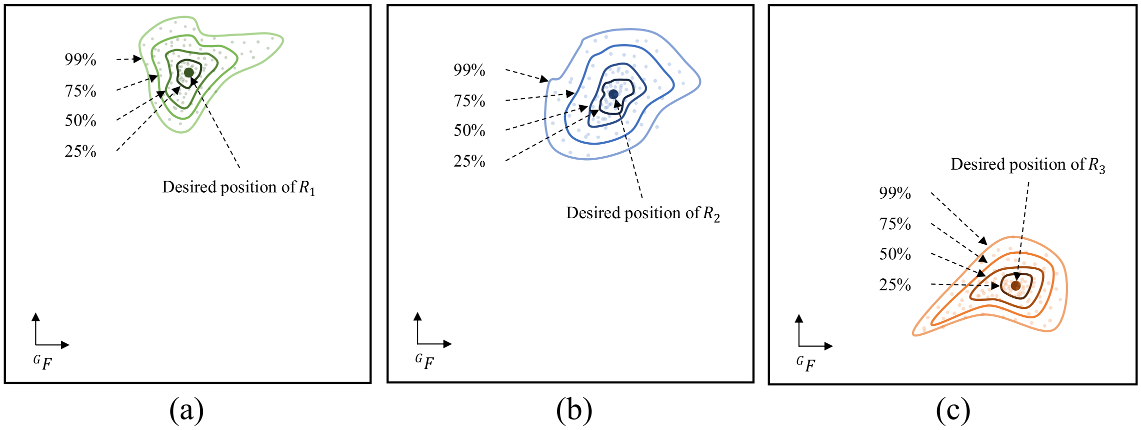


Fig. 6 Iso-probability curves for (**a**) R_1, (**b**) R_2, and (**c**) R_3 for the example shown in Fig. 2

4.1.2 Lost-Robot Grouping Criterion

Herein, it is conjectured that search redundancy can be minimized by grouping lost robots, whose RoIs overlap sufficiently (*i.e.*, merging their respective RoIs). In this regard, let A_{R_i} represent the area of the RoI of lost Robot R_i. Then, let $A_{R_iR_j}$ represent the overlapping area of the RoIs of the lost Robots R_i and R_j, respectively.

As shown in Fig. 7, for the example swarm disconnectivity in Fig. 2, the RoIs of the lost Robots (represented by their 99% iso-probability curves, respectively) R_1 and R_2 sufficiently overlap and, thus, these two robots may be grouped, if deemed beneficial, and their respective RoIs merged.

It is proposed, herein, that two lost robots R_i and R_j may be grouped, if their overlapping area, with respect to the maximum area of their respective RoIs, $M_{R_iR_j}$, exceeds a minimum threshold, M_{min}:

$$M_{R_iR_j} = \frac{A_{R_iR_j}}{\max\left(A_{R_i}, A_{R_j}\right)}. \tag{9}$$

Above, the threshold M_{min}, $0 < M_{min} \leq 1$, is a user-specified parameter.

4.1.3 RoI Merging

The possible RoI merging process starts by, first, calculating the overlap metric, $M_{R_iR_j}$, for all pairs of lost robots in the swarm, and identifying the pairs of lost robots whose overlap metric exceeds the user-defined threshold, M_{min}. Pairs of robots are grouped, if their $M_{R_iR_j}$ exceeds the set threshold. One must note, however, that this process may result in groups with more than just two lost robots, as well as, potentially, a lost robot belonging to multiple groups.

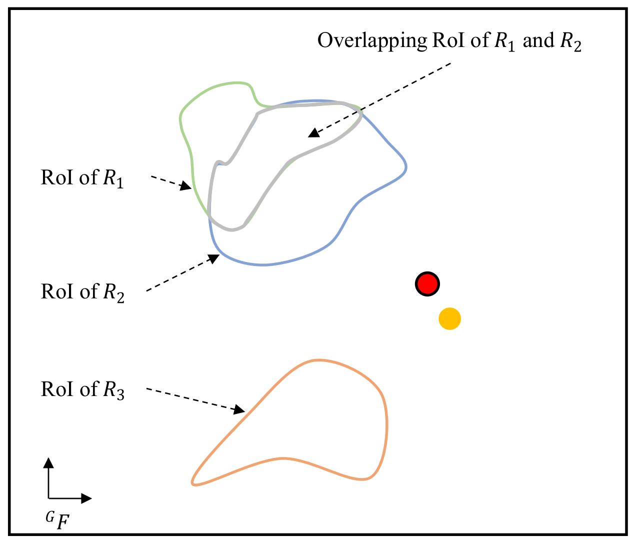


Fig. 7 Overlapping of RoIs for the example shown in Fig. 2

When lost robots are grouped, their corresponding merged RoI would be determined for the combined cloud of all points (*i.e.*, all possible considered probabilistic robot positions). The bounding iso-probability curve, p_u%, would be centered at the Cartesian positional mean of all (cloud) points.

The scalability of the RoI merging step can be improved by only merging the lost robots that are expected to be neighbors: Namely, only calculating the overlapping region of lost robots that are neighbors of each other, instead of calculating the overlapping region of all lost robot pairs. The expected neighbors of a lost robot can be determined based on

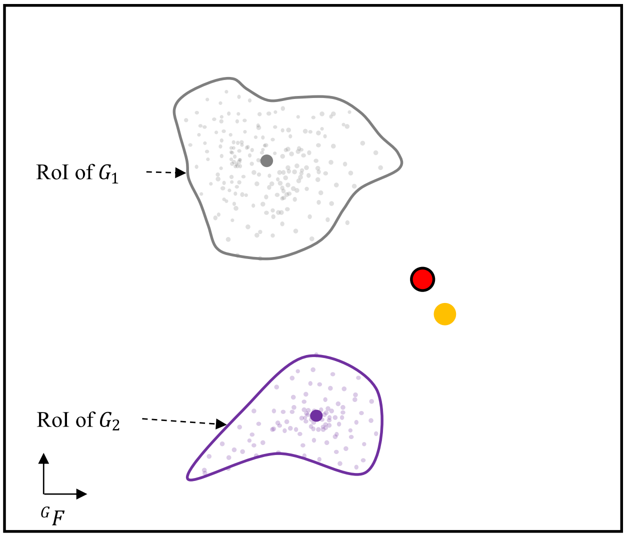


Fig. 8 RoIs of the lost-robot groups

their positions in the desired swarm configuration and a user-defined range.

Furthermore, the computational requirements of inter-region and intra-region path planning, detailed in Sections 4.2 and 4.3, respectively, depend on the number of regions that must be explored by the leader robot. The scalability of these two stages can, thus, be improved by selecting a lower minimum merging threshold, M_{min}.

Figure 8 shows the merged RoI of the lost robot group $G_1 = \{R_1, R_2\}$, as well as that of $G_2 = \{R_3\}$.

4.2 Inter-region Search Path Planning – Solving the Travelling-Salesperson Problem

Inter-region path planning aims at selecting the optimal order in which the RoIs would be searched by the leader robot. The optimal inter-region path (*i.e.*, represented by a *string*), $\boldsymbol{P}^*$, is determined herein by minimizing the objective function formulated above in Eq. (2), which considers the total distance that would need to be travelled by the searcher robot and the total search effort that would need to be expended to explore the RoI at hand:

$$\min J(\boldsymbol{P}) = \alpha \frac{D(\boldsymbol{P})}{D_{max}} + \beta \frac{C(\boldsymbol{P})}{C_{max}}. \tag{10}$$

Above, the total distance traveled by the searcher robot, $D(\boldsymbol{P})$, can be simply calculated as the sum of the Euclidean distances between the centroids of the RoIs on the inter-region path:

$$D(\boldsymbol{P}) = \sum\nolimits_{i=1}^{n_p} d(P_i), \tag{11}$$

where $d(P_i)$ is the distance between the i^{th} RoI, P_i, and the $(i-1)^{th}$ RoI, P_{i-1}.

The total search effort expended, on the other hand, can be calculated as the sum of the search efforts for all the RoIs on the inter-region path:

$$C(\boldsymbol{P}) = \sum\nolimits_{i=1}^{n_p} c(P_i), \tag{12}$$

where $c(P_i)$ is the search effort expended to explore the i^{th} RoI, P_i, calculated as the area of the RoI, $A_G(P_i)$, divided by the number of lost robots in the group associated with this region, $n_L(P_i)$:

$$c(P_i) = \frac{A_G(P_i)}{n_L(P_i)}. \tag{13}$$

One must note, however, the dynamic nature of the inter-region search-path planning optimization problem, Eq. (10), where RoIs may need to be reformed according to the sequence of RoIs, $\boldsymbol{P}$, examined. Namely, since some lost robots may belong to multiple groups, and that they would be detected during the exploration of the first/preceding group/RoI in the string that they belong to, they would not need to be considered again in a RoI to be examined down the line on the string, $\boldsymbol{P}$. On occasions, a RoI may even need to be removed completely from existence in a considered string, $\boldsymbol{P}$.

The abovementioned 'erasure' of lost robots from (future) RoIs cannot be carried out during the original lost-robot grouping stage without knowing the order of visitations of their respective RoIs. Thus, when calculating the objective function for a candidate inter-region path solution, $\boldsymbol{P}$, such scenarios must be taken into account. Furthermore, the optimal inter-region search path determined in this stage must pass down the optimal string, $\boldsymbol{P}^*$, while removing robots that are duplicated in the groups.

In essence, the inter-region path planning is a combinatoric optimization problem with a maximum of $n_{sol\boldsymbol{P}}$ solutions:

$$n_{sol\boldsymbol{P}} = perm(n_G, n_G), \tag{14}$$

where n_G is the number of RoIs, and *perm* is the permutation operator. The search space can be explored through any combinatoric search engine, such as Genetic Algorithms [67].

4.3 Intra-region Search Path Planning – Using Probability Theory for Finding Lost Robots

Intra-region path planning comprises the generation of an optimal path for the searcher robot to explore a RoI. If newly detected robots, in this RoI, were to be recruited to help the leader robot in its search for the remaining lost robots in the group, the overall team of searchers would be expected to maintain their formation, following detections, as they continue to move along the planned intra-region path.

The intra-region path generation method, proposed herein, maximizes the probability of detecting/ finding lost robots in a RoI. Our prior work in the context of wilderness search and rescue have examined the use of iso-probability curves for conducting searches for missing people [59–66]. In particular, the equal-effort search method presented in [64] is of interest herein. It equates the time and resources spent searching an area to the likelihood that the target will be in that area, Eq. (8) above:

$$p(t) \propto E(t). \tag{15}$$

Herein, we propose a bi-directional variation of the equal-effort search method. This search method would allow for spiral paths to be generated from outside of the RoI towards the center, as well as from the center outwards. Each generated path would cover a range of iso-probability curve percentiles defined by an upper and a lower percentile bound, p_u and p_l, respectively. One may choose p_l sufficiently small, based on the search team's detection range, for a near-exhaustive search.

In our proposed strategy, thus, the intra-region search path begins from the accurately known current position of the leader robot, travels along the shortest path (*i.e.*, initial path) toward the upper percentile bound of the next RoI in the string, $\boldsymbol{P}$, and, then, spirals inward toward the lower percentile bound, and subsequently spirals back outward toward the upper percentile bound. If necessary, this spiralling inward-outward pattern is repeated until the exploration of the RoI is completed. The intra-region search path generation is detailed in Appendix B.

Figure 9 illustrates an example intra-region search path planned for exploring the RoI of group G_1, where the leader robot spirals inward and outward only once for simplicity of illustration.

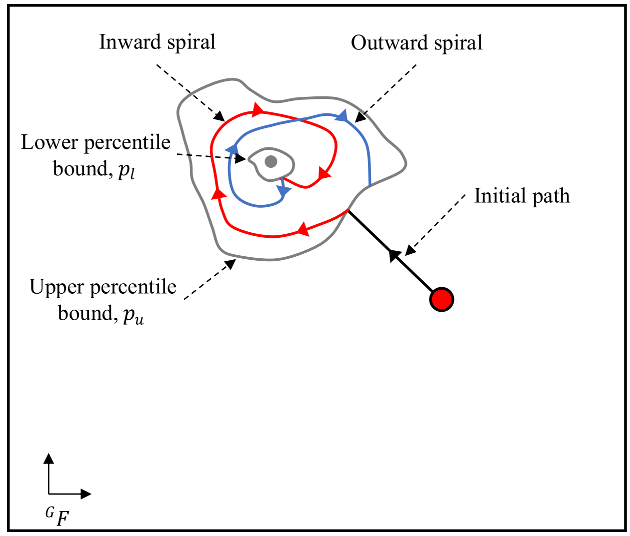


Fig. 9 An intra-region search path planned for exploring G_1 for the example disconnectivity in Fig. 2

4.4 Search Execution

Herein, a detailed description of the proposed connectivity restoration strategy is summarized in algorithmic form, as further shown in Fig. 10. All computations for RoI identification and inter/intra-region path planning are completed by the leader robot, thus, eliminating the need for global connectivity with an external base.

Once the swarm moves to its desired configuration, check connectivity:

1. If swarm is connected, exit the process.
   Otherwise, continue.
2. If robots that are connected to the leader exist, guide them to their respective locations in C_d.
   Otherwise, continue.
3. Identify the RoIs by grouping the lost robots through the methodology detailed in Section 4.1.
4. Determine the optimal inter-region search path through the methodology detailed in Section 4.2.
   Set iteration number $i = 1$ and start the inter-region search.
5. Determine the optimal intra-region search path, for exploring the RoI of P_i, through the methodology detailed in Section 4.3.
6. Execute the intra-region search path until a lost robot is detected. Once a lost robot is detected, continue.
   Note:

   (i) The intra-region search path is executed without cumulative motion errors. Namely, it is assumed that continuous connectivity between the recruited detected lost robots and the leader robot is feasible, limited by n_{Smax} (see notes for Step 7 for definition of n_{Smax}).

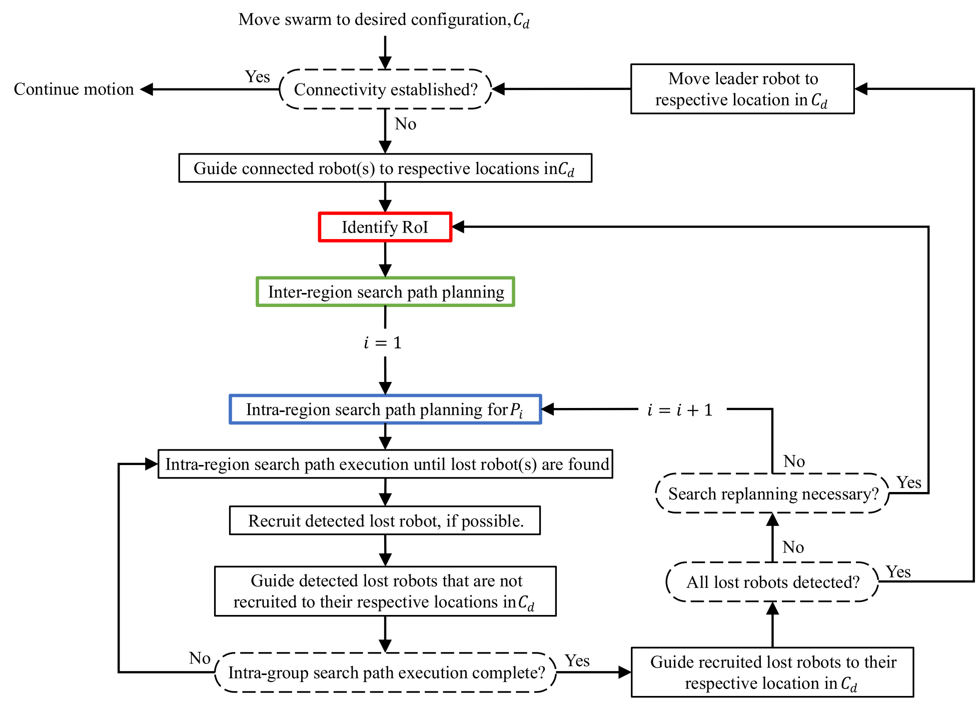


Fig. 10 Proposed connectivity restoration strategy flowchart

(ii) A detected lost robot may or may not be part of the lost robot group belong to the RoI at hand, P_i
.

(iii) The detection of a lost robot may lead to the detection of multiple other lost robots due to existing possible connectivity between them.

7. Recruit a maximum of n_{Smax} (detected) lost robots to help the leader in searching for the remaining lost robots in the RoI at hand.

Note:

(i) Due to the difficulty of continuous localization with an increased number of robots, only the first n_{Smax} detected robots join the leader robot in the search. n_{Smax} would be a user-defined value selected based on the communication capabilities of the member robots.

(ii) Recruited robots maintain their formation at the point of detection for the remainder of the search.

8. Guide the detected lost robot(s) that have not been recruited for search to their respective locations in the desired swarm configuration, C_d.

9. If execution of the intra-region path for lost robots in the region at hand, P_i, is not completed, return to Step 6.

Note: The intra-region path is considered completed only if:

(i) All lost robots in the group have been detected, or

(ii) The planned intra-region path is executed fully, though there still remain lost robots in the group corresponding to the region at hand that have not been detected. In this case, the lost robots that were not detected, for example due to failure of onboard communication, sensing, or locomotion, are considered as permanently disconnected. These robots are removed from the swarm.

The user may address the issue of *permanently disconnected robots* by, for example, reconfiguring the swarm (*i.e.,* selecting a new desired configuration without the permanently disconnected robot), after all remaining lost robots have been detected.

Otherwise, continue.

10. Guide the detected lost robots that were recruited for search to their respective locations in the desired configuration, C_d.

11. If all lost robots in the swarm have been detected, guide the leader robot to its respective position in the desired configuration, C_d, and return to Step 1.
    Otherwise, continue.
12. If search replanning is necessary, return to Step 3 to re-identify the RoI and re-plan the inter-region path.
    Otherwise, set $i = i + 1$, and return to Step 5.
    Note: The search must be replanned if the execution of the intra-region path led to the unintentional detection of lost that were not in the group corresponding to the RoI at hand.
    Note: If the proposed strategy returns to Step 3 for replanning, a simplified approach to inter-region path planning can be adopted to reduce the computational demands of the proposed strategy, as will be discussed below.

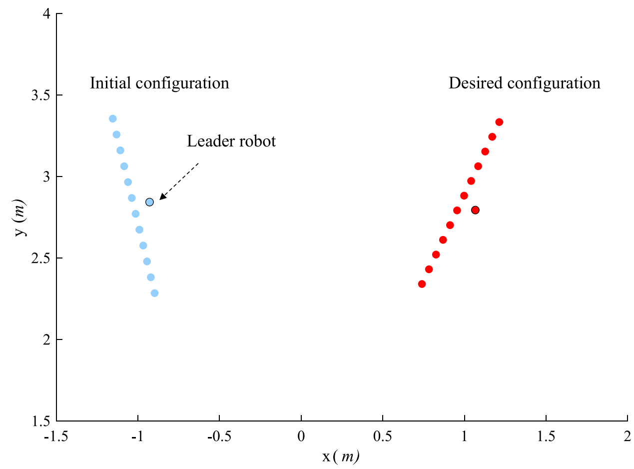


Fig. 11 Desired swarm motion

Simplified Inter-Region Search Path Planning The proposed connectivity-restoration strategy may require the inter-region search path to be replanned during the search due to the unintentional detection of lost robots that were not in the group corresponding to the RoI being explored. In such a case, a simplified approach to inter-region search path planning can be adopted.

The unintentional detection of a lost robot would result in re-identification of the RoIs (*i.e.,* grouping of the lost robots). Re-grouping would, however, not form groups additional to those that were formed in the previous invocation of the lost-robot grouping methodology. Namely, when a lost robot is unintentionally detected, it would be removed from all lost robot groups it is a part of, and potentially result in the elimination of entire lost-robot groups/RoIs. In this situation, it is proposed just to continue the original (optimal) inter-region path at hand by simply eliminating these regions.

5 Simulated Experiments

A series of simulated experiments were conducted to validate the effectiveness of the proposed connectivity restoration strategy. One of these experiments is detailed in Section 5.1. Subsequently, Section 5.2 provides a numerical comparison of our method to the competing baseline random-walk method.

5.1 An Illustrative Example

A numerical example of the proposed connectivity-restoration strategy is presented for a swarm of thirteen (twelve followers plus a leader) robots moving from an initial configuration to a desired one, Fig. 11. The connectivity of the swarm is (only) checked once the swarm reaches the desired configuration.

In this example, due to motion errors, connectivity is assumed to be lost once the swarm arrives at its desired configuration, Fig. 12. The proposed connectivity-restoration strategy is, thus, invoked to move the robots to positions that re-establish the desired connectivity.

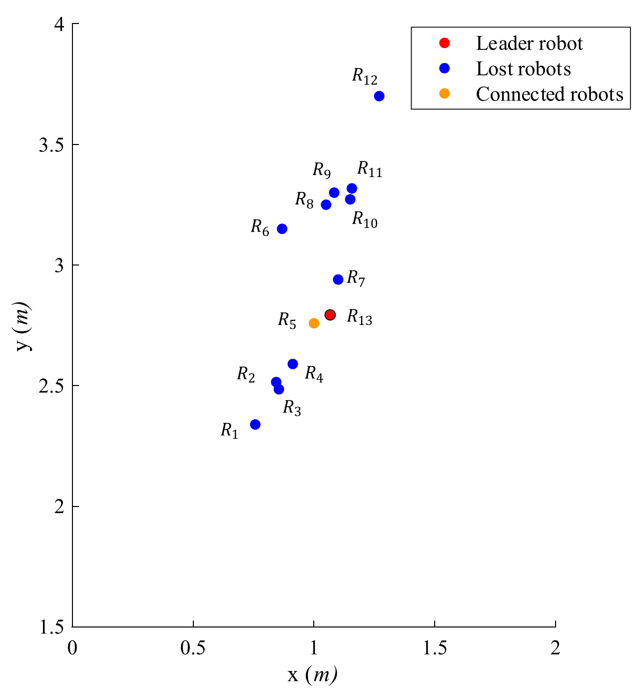


Fig. 12 Swarm configuration upon reaching its desired destination – in a disconnected state

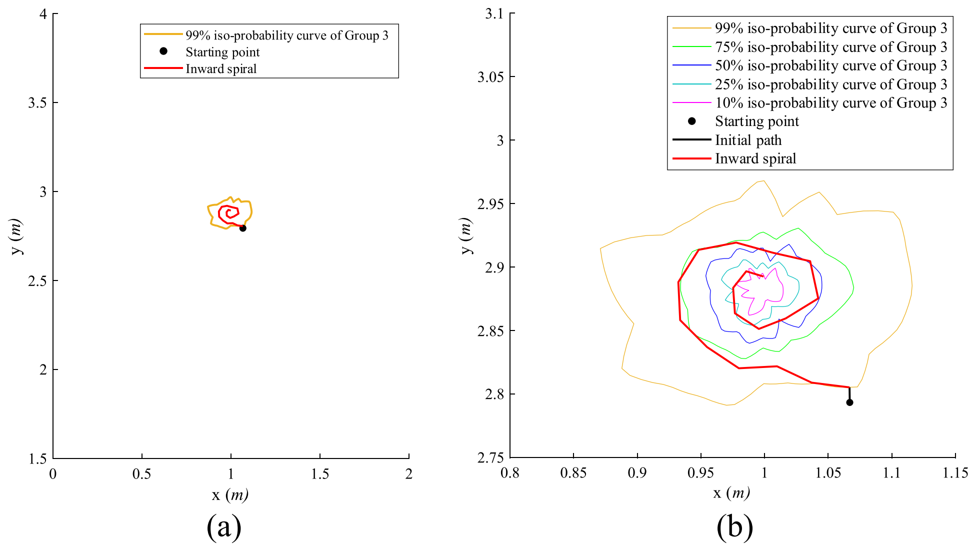


Fig. 13 (**a**) Intra-region path for exploring the RoI of G_3, (**b**) five example iso-probability curves

As the first step of the restoration process, the *connected* robot, R_5, is guided to its respective location in the desired configuration. This initial motion is carried out in parallel to the execution of the intra-region path for exploring the first RoI on the planned inter-region search path, Section 5.1.4 below. This is achieved by localizing it based on the (accurate) position of the leader robot and inter-robot proximity measurements obtained using onboard hardware. Such measurements describe the relative distance and bearing between two neighboring robots.

Once R_5 reaches its destination it may be disconnected from the leader robot. However, we still consider R_5 as a *connected* robot that was sent to its destination and, thus, do not check for swarm connectivity until all lost robots are found and sent to their respective positions on the desired swarm configuration, as detailed above in Section 4.4.

5.1.1 Examining the Search Space – Identifying Regions-of-Interest

In this paper, the RoIs of the individual lost robots were represented through their respective $p_u = 99\%$ iso-probability curves. Some were, subsequently, merged using a threshold of $M_{min} = 0.5$, Eq. (9) in Section 4.1, resulting in the following $n_G = 7$ lost-robot groups:

$$\begin{aligned} G_1 &= \{R_1\}, \\ G_2 &= \{R_6\}, \\ G_3 &= \{R_7\}, \\ G_4 &= \{R_2, R_3\}, \\ G_5 &= \{R_3, R_4\}, \\ G_6 &= \{R_8, R_9\}, \text{ and} \\ G_7 &= \{R_{10}, R_{11}, R_{12}\}. \end{aligned} \tag{16}$$

Above, the lost Robot R_3 belongs to two groups: G_4 and G_5.

5.1.2 Inter-region Search Path Planning – Solving the Travelling-Salesperson Problem

The inter-region search path planning step determines the optimal order in which the RoIs should be visited, Section 4.2. Herein, for Eq. (10) the weights were chosen as $\alpha = \beta = 0.5$, which yielded the following (optimal) inter-region search path:

$$\boldsymbol{P}^* = \{G_3, G_6, G_7, G_2, G_5, G_4, G_1\}. \tag{17}$$

5.1.3 Intra-region Search Path Planning – Using Probability Theory for Finding Lost Robots

As noted above, the inter-region search path, $\boldsymbol{P}^*$, execution begins by exploring the RoI of the lost-Robot R_7, in Group G_3 through a planned intra-region path, Section 4.3. Upper and lower bounds of $p_u = 99\%$ and $p_l = 10\%$ iso-probability curves were used to plan this path.

Figure 13(a) shows the intra-region path planned for the RoI of G_3. Figure 13(b) further shows the 10%, 25%, 50%, 75% and 99% iso-probability curves as a zoomed-in illustration. Only a few example iso-probability curves and one inward spiral are shown for simplicity.

5.1.4 Search Execution

Below, the entire search simulation is discussed in detail.

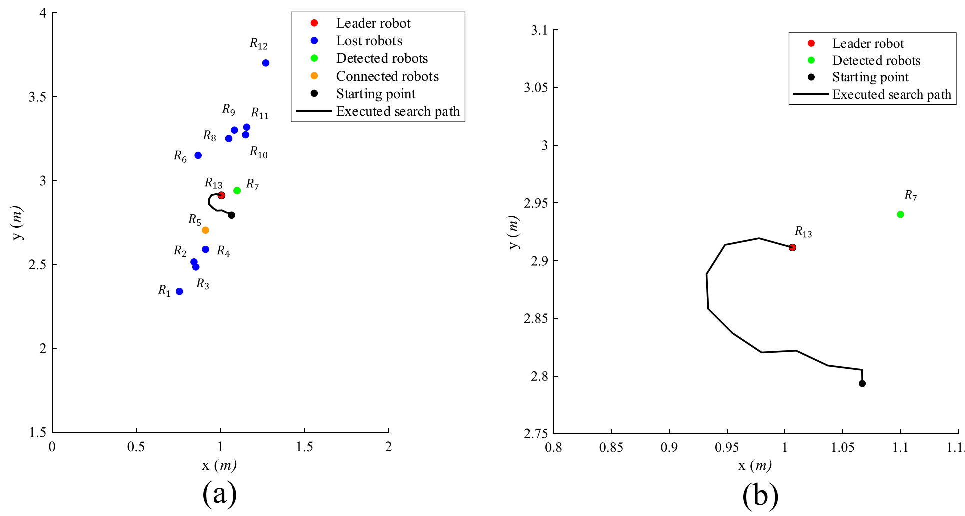


Fig. 14 Intra-region path for the detection of lost robots in G_3

(a) Search for the lost robots in $G_3 = \{R_7\}$

As the first step of the search, the detection of Robot R_7 by the leader robot is shown in Fig. 14(*a*), and a zoomed-in illustration of the executed path is shown in Fig. 14(*b*). The execution of the intra-region search for G_3 was considered complete once Robot R_7 was detected. Subsequently, R_7 was labeled as a *connected* robot and guided to its respective position in the desired swarm configuration. The leader robot, then, proceeded to explore the remaining RoIs on its optimal inter-region path, Eq. (17).

(b) Search for the lost robots in $G_6 = \{R_8, R_9\}$

As the next search step, Eq. (17), the intra-region path planned to search for the robots in Group G_6 is shown in Fig. 15(a). The execution of this path led to the simultaneous detection of lost Robots R_8, R_9, R_{10} and R_{11}, Fig. 15(b). Namely, Robot R_8 was first detected directly by the leader robot, and the other lost robots, R_9, R_{10}, and R_{11}, were detected indirectly due to existing, however, a priori unknown, connectivity between them and R_8. All newly detected/found lost Robots, $R_8 - R_{11}$ were designated as *connected* and guided to their respective positions in the desired swarm configuration.

(c) Replanning

As noted above, the execution of the intra-region path for the RoI of G_6 led to the *unintentional* detection of R_{10} and R_{11}, which were not part of G_6 and as such, the connectivity-restoration process had to be replanned. As detailed in Section 4.4, replanning removed R_{10} and R_{11} from G_7 and yielded the following new optimal inter-region path:

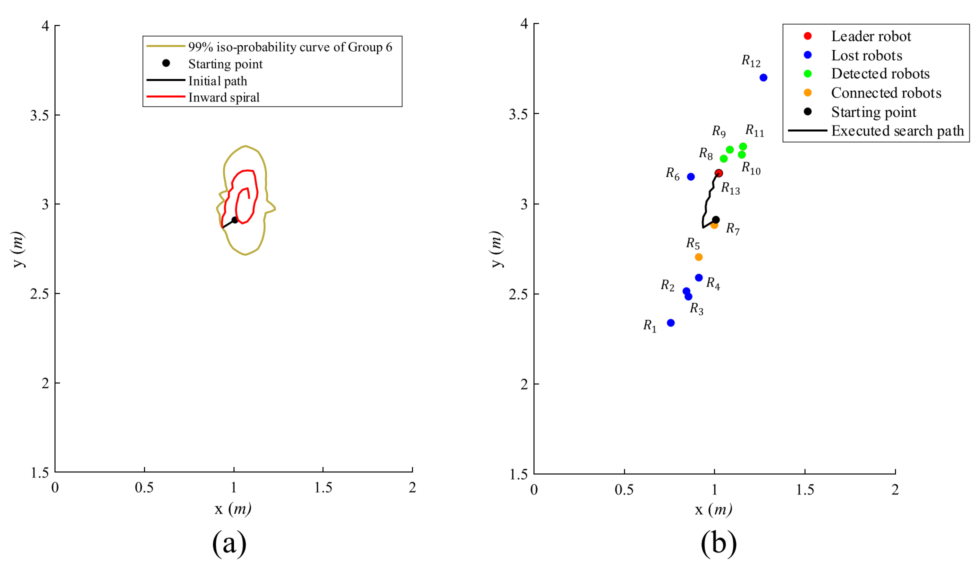


Fig. 15 (**a**) Planned and (**b**) executed intra-region path for the detection of lost robots in G_6

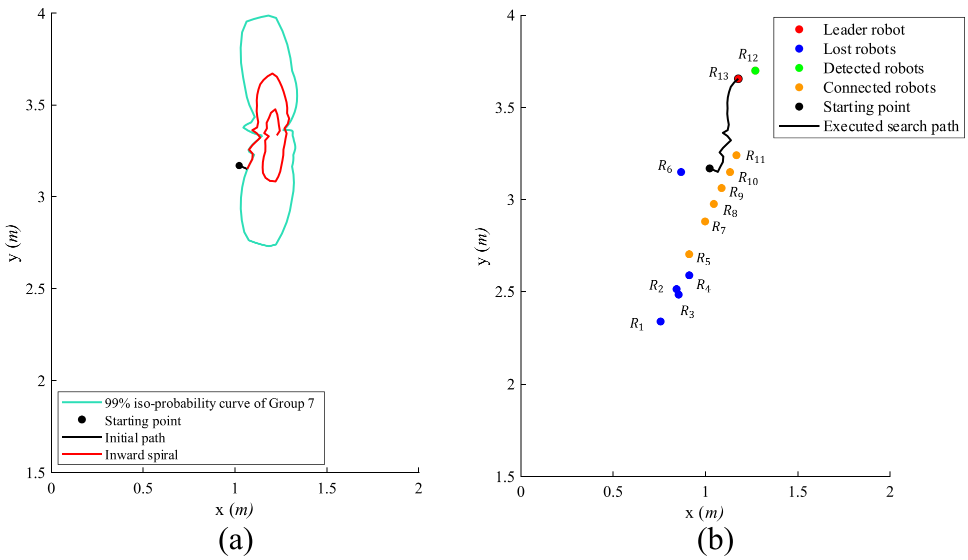


Fig. 16 (**a**) Planned and (**b**) executed intra-region path for the detection of lost robots in G_7

$$P^* = \{G_7, G_2, G_5, G_4, G_1\}, \tag{18}$$

where,

$$\begin{aligned} G_1 &= \{R_1\}, \\ G_2 &= \{R_6\}, \\ G_4 &= \{R_2\}, \\ G_5 &= \{R_3, R_4\}, \text{ and} \\ G_7 &= \{R_{12}\}. \end{aligned} \tag{19}$$

(d) Search for the lost robots in new $G_7 = \{R_{12}\}$

The restoration process continued via the exploration of the RoI of Group G_7, whose planned intra-region path is shown in Fig. 16(a). The execution of this path led to the detection of Robot R_{12}, Fig. 16(b). Subsequently, R_{12} was labeled as a *connected* robot and guided to its respective position in the desired swarm configuration, Section 4.4. The leader robot, then, proceeded to explore the remaining RoIs on its optimal inter-region path, Eq. (18).

(e) Search for the lost robots in $G_2 = \{R_6\}$

As the next search step, Eq. (18), the restoration process continued by the exploration of the RoI of Group G_2, for Robot R_6, whose planned intra-region path is shown in Fig. 17(a). The execution of this path, however, first led to the detection of lost Robots R_2, R_3, and R_4, which are members of Groups G_4 and G_5, respectively. Namely, these were

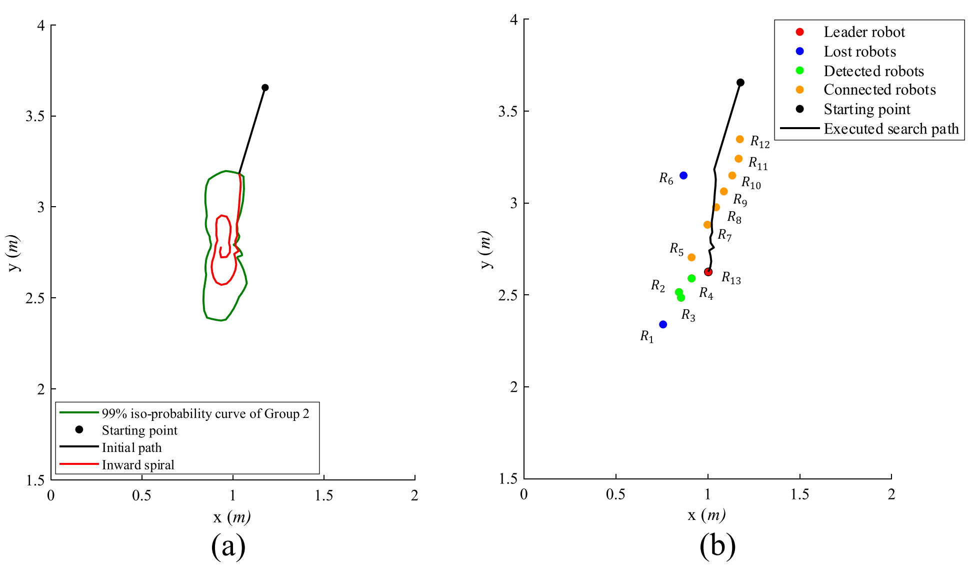


Fig. 17 (**a**) Planned and (**b**) executed intra-region path for the detection of lost robots in G_2

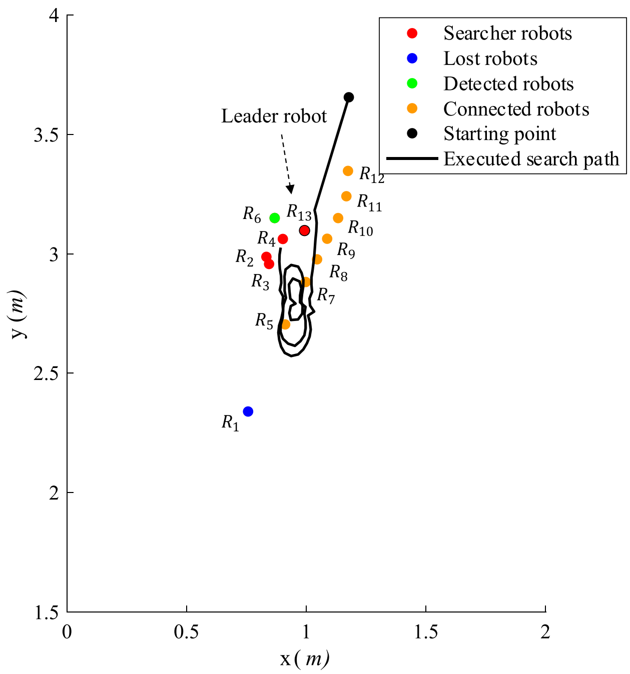


Fig. 18 Phase two of intra-region path execution for the detection of lost robots in G_2

found unintentionally. Robot R_4 was detected directly by the leader robot, and the two other lost robots, R_2 and R_3, were detected indirectly due to existing, however, a priori unknown, connectivity between them and R_4, Fig. 17(b).

All newly detected robots were, subsequently, recruited to help the leader robot with its search for the actual intended target in G_2, the lost Robot R_6, executing the originally planned intra-region path. This recruitment was allowed as the intended target, R_6, was yet to be detected. The newly formed search team eventually detected Robot R_6, Fig. 18, which completed the intra-region search at hand. All detected robots, R_2, R_3, R_4, and R_6, were, then, labelled as connected and guided to their respective positions in the desired swarm configuration.

(f) Replanning

The lost Robots R_2, R_3, and R_4, that were detected in the previous step did not belong to the RoI/group at hand, G_2. Thus, the restoration process had to be replanned, as detailed in Section 4.4, yielding the following new (final) optimal inter-region path:

$$\boldsymbol{P}^* = \{G_1\}, \tag{20}$$

where

$$G_1 = \{R_1\}. \tag{21}$$

(g) Search for the lost robots in new $G_1 = \{R_1\}$

The new planned intra-region path, Eq. (20), for exploring the new RoI of G_1 is shown in Fig. 19(a). The execution of this path detected the final lost Robot R_1, Fig. 19(b), who was labeled as connected and guided to its position in the desired swarm configuration.

(h) Guiding the leader robot to its desired position

Once all lost robots were detected/found, the leader robot also returned to its position in the desired swarm configuration and swarm connectivity was rechecked. Figure 20

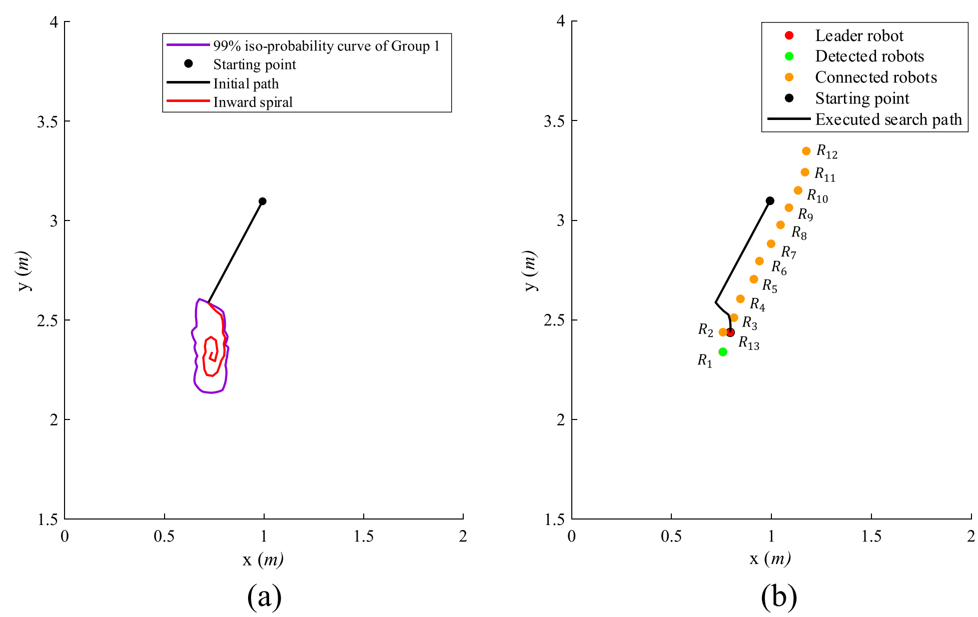


Fig. 19 (**a**) Planned and (**b**) executed intra-region path for the detection of lost robots in G_1

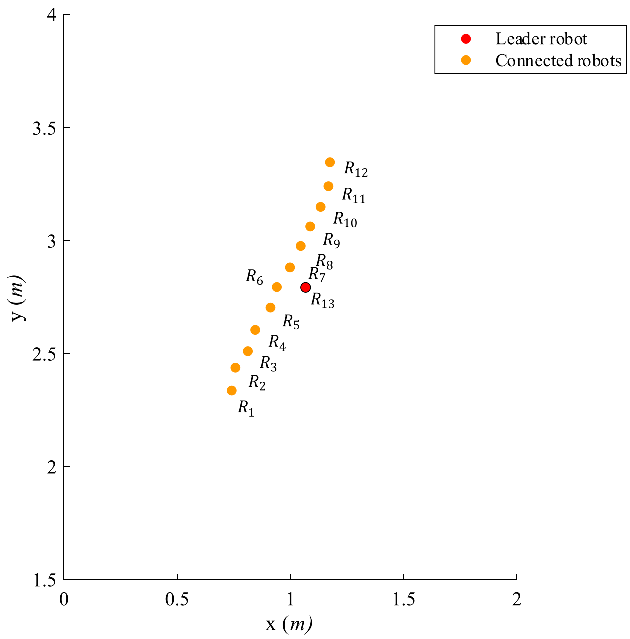


Fig. 20 Restored swarm connectivity

illustrates the positions of all swarm robots once the leader reaches its destination. In this example, the connectivity of the swarm was successfully restored. If connectivity were not to be established, the restoration process would need to be repeated from this point.

5.2 Comparison to Random-Walk Search Method

The proposed connectivity restoration strategy is compared to a competing *random-walk search strategy*, via the example presented in Section 5.1, for the disconnected swarm shown in Fig. 12 above. The random-walk strategy is first detailed in Section 5.2.1, and its performance compared to the proposed connectivity restoration strategy in Section 5.2.2.

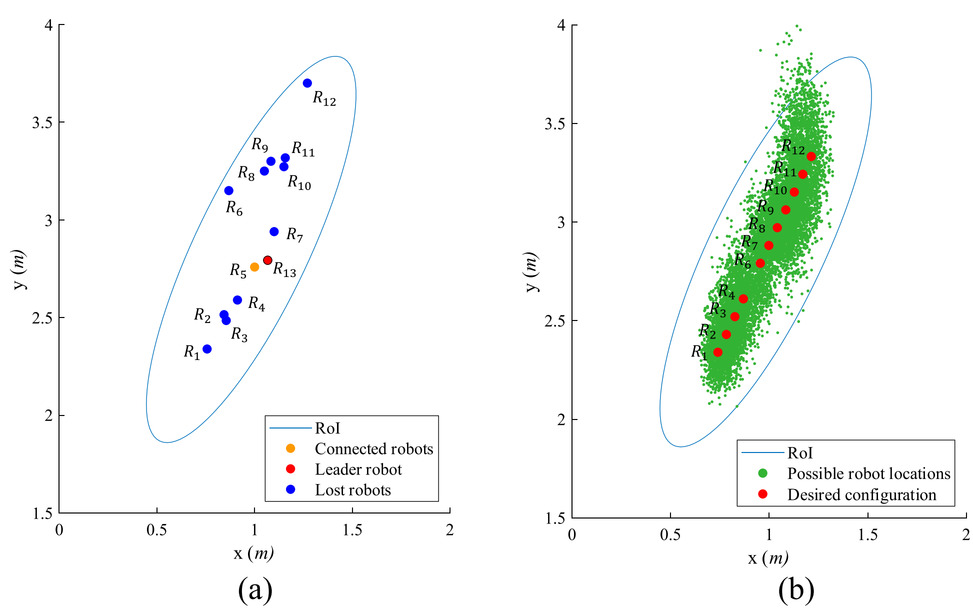
A random-walk strategy was selected for comparison as it is the only method in the swarm connectivity-restoration literature that is applicable to the problem at hand. As detailed in Section 2, existing strategies, typically, require the position of the lost robots to be known with certainty, and/or require the lost robots to navigate their environment without the accumulation of random errors. Neither of these conditions can be met in the problem considered in this paper. Behavior-based methods can be applied to the leader robot while the lost followers remain stationary. Of these methods, backtracking to the last position where the swarm was connected [42, 43] and random walk [44] were considered. The former strategy was, however, disregarded as it would not allow the leader robot to explore the RoIs in their entirety.

5.2.1 Random-Walk Search Strategy

In the random-walk search strategy, the leader robot explores its surrounding by executing a random-walk within a *bounding region-of-interest*. Herein, this bounding region is selected as an ellipse, centered at the centroid of the (potential) positions of the lost robots in the desired swarm configuration, Fig. 21. The selected RoI encircles 99% of the points in the cloud that represents the estimated deviations of all lost robots from their desired positions.

The random-walk search strategy is detailed below and illustrated in Fig. 22:

Fig. 21 (**a**) Disconnected swarm, and (**b**) corresponding RoI for the random-walk search strategy



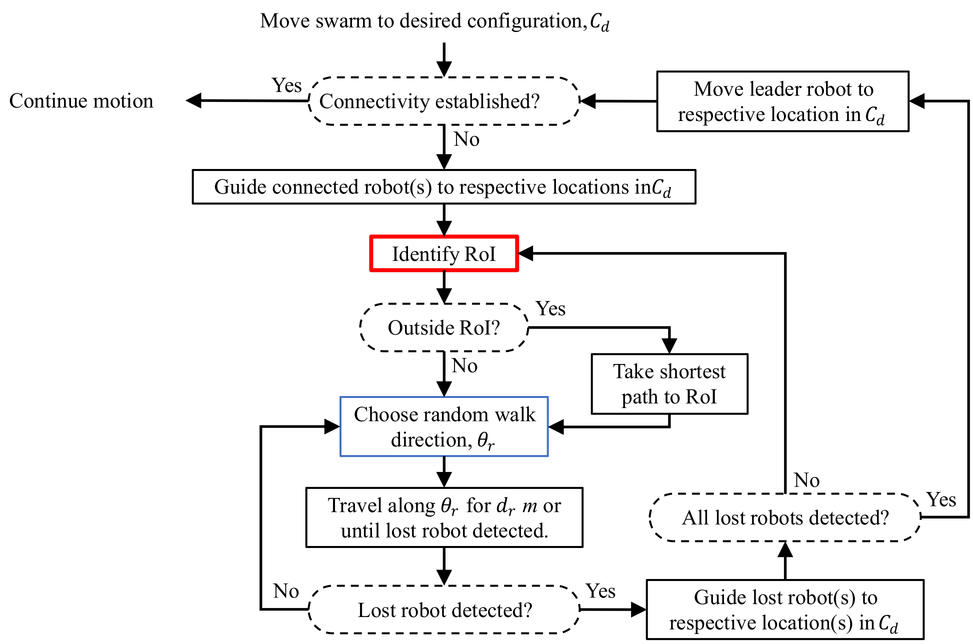


Fig. 22 Random-walk search strategy flowchart

1. If swarm is connected, exit the process.
   Otherwise, continue.
2. If robots that are connected to the leader exist, guide them to their respective locations in C_d.
   Otherwise, continue.
3. Identify the bounding RoI based on the approach detailed above.
4. If leader robot is outside bounding RoI, take the shortest path to reach this RoI.
   Otherwise, continue.
5. Choose a random-walk direction, θ_r, based on a uniform distribution (*i.e.,* $\theta_r \sim U(0, 2\pi)$).
6. Travel along a randomly selected direction, θ_r, for a distance of d_r or until a lost robot is detected.
   Note: The distance d_r is a user-defined parameter.
7. If a lost robot is not detected, return to Step 5.
   Otherwise, continue.
8. Guide the detected lost robots to their respective locations in the desired configuration, C_d.
9. If all lost robots in the swarm have been detected, guide the leader robot to its respective position in the desired configuration, C_d, and return to Step 1.
   Otherwise, return to Step 3, to identify a new bounding RoI for the remaining lost robots.

5.2.2 Numerical Comparison

Due to the random nature of the random-walk search strategy, its performance in restoring swarm connectivity must be evaluated over multiple repeated simulations. Furthermore, one must also specifically note the impact of the value of the walk distance, d_r, on the strategy's performance. In this regard, numerous sets of simulations were conducted, first, to evaluate the impact of d_r, in the range 0.1 to 0.9 m. The results clearly showed that there indeed exists an optimal value for the walk distance, d_r^*, 0.5 m in our case, for the disconnectivity scenario examined. During all the simulations, the direction of the walk was selected using a uniform distribution, in the range 0 to 2 π radians.

The example shown in Fig. 23 illustrates an example path taken by the leader robot while using the random-walk search strategy with $d_r = 0.5m$. The solution is dynamic in nature. Namely, the RoI at hand is recalculated, if needed, every time a lost robot is found – its size is reduced by removing the points that belong to the newly found robot from the 'cloud' at hand.

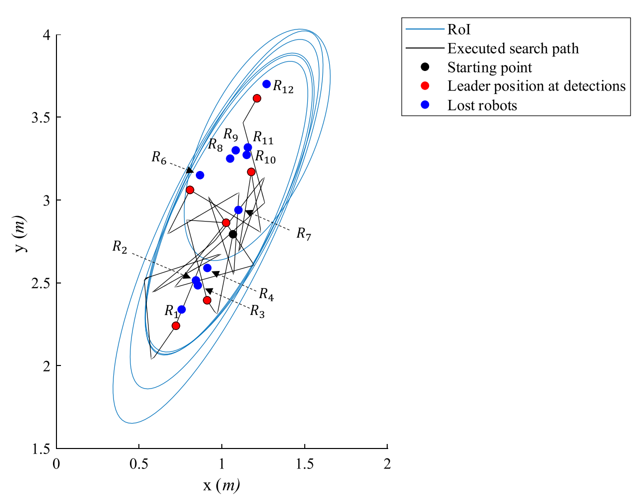


Fig. 23 Random-walk search strategy

Table 1 Connectivity restoration performance

Method	Proposed Method	Random-walk, $\boldsymbol{d}_r^* = 0.5$ m		Random-walk, $\boldsymbol{d}_r = 0.1$ m	
Total distance travelled, $\boldsymbol{d}_S$	4.9 m	Min	Max	Min	Max
		3.3 m	59.1 m	7.2 m	123.3 m

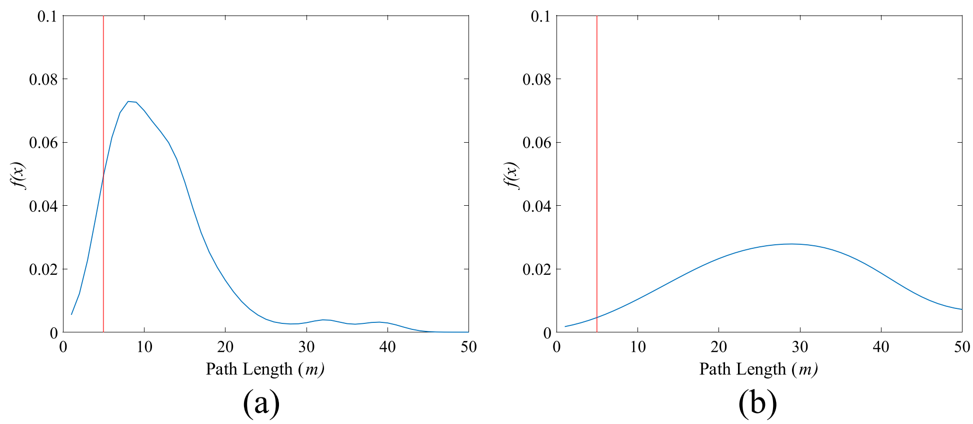


Fig. 24 Performance of the random-walk search strategy for (**a**) $d_r^* = 0.5\,m$, and (**b**) for $d_r = 0.1\,m$

The performances of the two competing search methods are presented in Table 1. For the random-walk strategy, two walk distances are included: 0.1 m and 0.5 m, the latter being the optimal value for the example considered. For each value, 100 (random) simulations were implemented. Search performance was evaluated based on the total distance travelled by the leader robot, d_S, from its initial position once the swarm entered a disconnected state, until all lost robots were found.

Figure 24(a), below, illustrates the distribution of the results for the 100 instances of the random-walk search strategy with $d_r^* = 0.5\,m$. The performance of the proposed probabilistic connectivity restoration strategy is also shown as a vertical red line. For this example, the proposed probabilistic connectivity-restoration strategy outperformed the random-walk strategy in approximately 85% of instances, indicating the tangible efficacy of our strategy. Furthermore, however, it would be infeasible to conduct an optimization to determine d_r^*. Thus, for a randomly chosen d_r metric, for example 0.1 m in Table 1, the value of our proposed method over the random-walk method could be even significantly better, Fig. 24(b).

It must be noted that the time taken to restore connectivity may need to consider issues beyond robot travel time, including (1) inter-robot communication and (2) search replanning. Namely, due to latency and bandwidth limitations, there may be a delay in the inter-robot communication necessary for detecting lost robots, recruiting them, and/or guiding them to their destinations. Such limitations would depend on the communication capabilities of the member robots. Furthermore, search replanning may be a time-consuming process due to the limited onboard computational capabilities of the leader robot. Such metrics were, however, not considered in this work for the purpose of simplifying the discussion and remaining platform independent.

6 Conclusions

This paper presents a connectivity-restoration strategy for robotic swarms that lose connectivity due to random errors in the execution of their motion commands. Connectivity restoration is particularly challenging in such a scenario as motion-execution errors must be considered in the restoration strategy, and the position of the disconnected robots must be probabilistically modelled.

The proposed connectivity-restoration strategy addresses the former challenge by maintaining the disconnected lost robots stationary until they are detected by the leader robot that is equipped with enhanced localization capabilities. The leader explores its surroundings by dividing the environment into multiple regions of interest (RoI), determining the optimal order of RoI visitations, and exploring each RoI based on a probabilistic model of the potential positions of the robots. Herein, iso-probability curves are used for this probabilistic representation, which allow the leader robot to prioritize exploring areas with a higher probability of lost-robot detection, consequently leading to reduced connectivity restoration time.

The strategy is novel as it uses probability theory to plan the overall search path of the leader robot. It is also novel as it may recruit lost robots that are detected during the search to form a search team for collectively exploring the environment for the remaining lost robots. This allows it to expedite the restoration time by increasing the area that can be detected at any one time. The efficacy of the proposed strategy was illustrated through a detailed simulated experiment for a swarm of thirteen robots engaged in a motion control task. The performance of the strategy was also compared to a competing random-walk search strategy through extensive simulations, showing its efficiency in restoring swarm connectivity.

Connectivity restoration is guaranteed, in our work, if (1) the leader explores all parts of the determined RoIs, and (2) the lost robots are located within these regions. Condition (1) is met by the proposed strategy through the intra-path planning stage, where a spiral path that covers all parts of the region at hand is planned. Condition (2), however, may not be met if the lost robots were to be located outside of the upper-bound iso-probability curves that define the RoIs. It is, thus, recommended to select a high value for the upper-bound iso-probability curve (*i.e.,* to increase the size of the RoI) to increase the probability of restoring connectivity.

While the proposed strategy was developed for a single-leader, multiple-follower swarm architecture, it could also be applicable to a distributed case where the swarm comprises multiple groups, each with its own leader robot, respectively. In this scenario, each leader robot would be responsible for restoring connectivity to the followers in its group. Furthermore, each leader would have to be assigned the set of followers that it is responsible for through an offline or online optimization process.

Appendix A. Determining Iso-Probability Curves

Iso-probability curves have been used in several robotic wilderness search and rescue (WiSAR) works to model the estimated location of a lost person [59–66]. Each iso-probability curve is associated with a percentile p and in the WiSAR context it describes how far in a given direction the p^{th} slowest percentile of the missing person would have moved at a given point in time. As p varies from 0 to 100%, the iso-probability curves provide a complete description of the missing person's possible locations. In a more general sense, each iso-probability curve describes how far, from the center of the (closed-loop) curve, one would need to travel to have a p percent chance that the target be between that location and the center. In this a way, set of iso-probability curves provides a probabilistic description of the possible locations of a search target in a way that can conveniently describe the area where the target might be and how to cover that area with different curves.

Mathematical Formulation

The formulation for iso-probability curves presented in [65] provides a more rigorous mathematical description. It defines an iso-probability curve for a time varying probability density function in polar coordinates, $\rho_p(r,\theta,t)$, as follows:

$$\left(F^{-1}(p|\theta,t),\theta\right)\forall\theta\in[0,2\pi], \tag{22}$$

where F^{-1} is the inverse cumulative probability density in a given direction θ at time t. When evaluated for a given quantile p, at all values of θ, this yields all points along the p^{th} iso-probability curve. To compute the desired inverse cumulative density function F^{-1}, both the cumulative density function F and the probability density function f are computed from the polar density function as follows:

$$F(r|\theta,t)=\int_0^r f(s|\theta,t)ds, \text{ and} \tag{23}$$

$$f(r|\theta,t)=\frac{\rho_p(r,\theta,t)}{\int_0^\infty \rho_p(s,\theta,t)ds}. \tag{24}$$

Estimation from Point Clouds

In this work, the estimation of robot position is to be estimated through a cloud of points, generated based on the robot's motion model and its last known position. This point cloud estimation of the lost-robot position must be converted to a probability density estimate for use with the iso-probability curve formulation above. The approach shown in [66] uses a polar grid that is equally spaced in the angular and radial components to construct a histogram-based density estimate. Each point in the point cloud is binned to the closest point in the grid and divided by the total number in that direction, providing an estimate of the probability density in each direction, f. Thereafter, the values at each point are summed in the radial direction providing an estimate of the cumulative density in each direction, F. Interpolation can, then, be used to provide the values of F^{-1} needed to construct each iso-probability curve.

Appendix B. Intra-Region Search-Path Planning

An intra-region search path is generated in this work over three steps. Firstly, the percentile (*i.e.,* iso-probability curve) that should be searched is determined. Next, the polar-coordinate position of the search path on that curve is calculated.

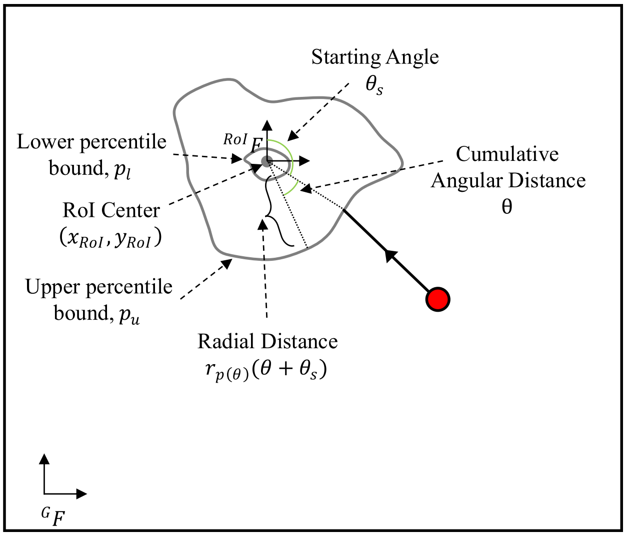


Fig. 25 Variables and measurements used for the determination of the intra-region path

Finally, the polar-coordinate position of the path is converted into Cartesian coordinates to determine the intra-region path. Figure 25 shows the key information that is extracted from the Region-of-Interest (RoI) for use in determining the search path.

Step 1: Percentile Selection

The search method used in this work generates spiral search paths by relating the percentile of the polar distribution that is being searched at a point on the path to the cumulative time and resources that have been used searching up to that point on the path. In this case, the cumulative angular distance that the path has traveled is used as a measure of the time and resources used while searching. Namely, as the spiral path progresses around the center of a RoI, it will follow lower or higher percentile iso-probability curves in a manner that is proportional to the number of turns the spiral has made.

Formally, let us consider the cumulative angular distance that has been traveled, $\theta \in [0, \theta_{max}]$. This cumulative distance can be measured in radians and would increase from 0, at the start point of the spiral search, to some maximum value, θ_{max}, that corresponds to the number of turns in the desired spiral path. For example, the first spiral path inwards, while making two complete turns, would have a cumulative angle ranging from 0 to 4π and the starting point where the cumulative angle is zero would be where the initial search path meets the iso-probability curve for the upper bound of the RoI. This cumulative angle is, then, converted into the percentile p being searched at that point on the path using a linear relationship:

$$p(\theta) = p_s + C\theta, \tag{25}$$

where p_s is starting percentile for the search, and C is a constant progression rate controlling how quickly the path moves across different iso-probability curves. The spiral paths are defined between a starting percentile and a final percentile. For paths that spiral outwards, the starting percentile is the lower bound (*i.e.,* $p_s = p_l$), the final percentile is the upper bound, and the progression rate is strictly positive (*i.e.,* $C > 0$). Conversely, for inwards spirals, the starting percentile is the upper bound (*i.e.,* $p_s = p_u$), the final percentile is the lower bound, and the progression rate is strictly negative (*i.e.,* $C < 0$). The percentile bounds can also be used to calculate the maximum cumulative angular distance using:

$$\theta_{max} = \frac{p_u - p_l}{abs(C)}. \tag{26}$$

Step 2: Polar-coordinates Path

Given the cumulative angle, θ, the starting angle of the path, θ_s, the percentile being searched at that point, $p(\theta)$, and the radial position of the p^{th} iso-probability curve at a given angle, $r_{p(\theta)}(\theta + \theta_s)$, one can define the complete search path. This can be achieved by evaluating $r_{p(\theta)}(\theta + \theta_s)$, the radial position of the iso-probability curve for the percentile determined by Eq. (B1) at θ, for values of θ along the path, resulting in a path in polar coordinates in the frame centered on the RoI being searched.

Step 3: Cartesian-coordinates Path

The polar-coordinates path would need to be converted into Cartesian coordinates in the global frame to determine the intra-region search path. This is achieved using the standard transformation between polar and Cartesian coordinates and the translation of the reference frame. Using the center point of the RoI being searched, (x_{RoI}, y_{RoI}), the following transformation is obtained:

$$x = x_{RoI} + r_{p(\theta)}(\theta + \theta_s)cos(\theta + \theta_s), \text{ and} \tag{27}$$

$$y = y_{RoI} + r_{p(\theta)}(\theta + \theta_s)sin(\theta + \theta_s). \tag{28}$$

The above transformation can be applied to arbitrary points along the path yielding the desired intra- region search path, S, as a collection of points in Cartesian-coordinates in the global frame that can be followed by the search team.

Acknowledgements This work was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC), and the Canada Research Chairs program (CRC).

Author Contributions Conceptualization: KE, DR, GN, and BB; Methodology: KE and NNS; Software: KE, NNS, and CH; Analysis: KE, NNS, and CH; Writing: KE, NNS, CH, GN, and BB.; Funding acquisition: GN and BB; Supervision: GN and BB. All authors read and approved the final manuscript.

Funding This work was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC), and the Canada Research Chairs program (CRC).

Declarations

Competing Interests Author Beno Benhabib is an Executive-Board Member for the Journal of Intelligent and Robotic Systems. The authors have no relevant funding, employment, financial interests to disclose.

Ethics approval Not Applicable.

Consent to Participate Not Applicable.

Consent to Publish Not Applicable.

References

1. Barca, J.C., Sekercioglu, Y.A.: Swarm robotics reviewed. Robotica **31**(3), 345–359 (2013)
2. Brambilla, M., Ferrante, E., Birattari, M., Dorigo, M.: Swarm robotics: A review from the swarm engineering perspective. Swarm Intell. **7**(1), 1–41 (2013)
3. Şahin, E.: Swarm robotics: From sources of inspiration to domains of application. In: Şahin, E. and Spears, W.M. (eds.) Swarm Robotics. pp. 10–20 (2004)
4. Schranz, M., Umlauft, M., Sende, M., Elmenreich, W.: Swarm robotic behaviors and current applications, Front. Robot. AI **7**, (2020)
5. Farid, M., Kamal, M.A.S., Egerton, S.: Search strategies and specifications in a swarm versus swarm context. Robotica **39**(11), 1909–1925 (2021)
6. Abdelli, A., Yachir, A., Amamra, A., Khaldi, B.: Maximum likelihood estimate sharing for collective perception in static environments for swarm robotics". Robotica **41**(9), 1–20 (2023)
7. Couceiro, M.S., Portugal, D., Rocha, R.P., Ferreira, N.M.: Marsupial teams of robots: deployment of miniature robots for swarm exploration under communication constraints. Robotica **32**(7), 1017–1038 (2014)
8. Stergiopoulos, Y., Tzes, A.: Decentralized swarm coordination: A combined coverage/connectivity approach. J. Intell. Robot. Syst. **64**, 603–623 (2011)
9. Eshaghi, K., Li, Y., Kashino, Z., Nejat, G., Benhabib, B.: mROBerTO 2.0 – An autonomous millirobot with enhanced locomotion for swarm robotics. Robot. Autom. Let **5**(2), 962–969 (2020)
10. Kim J.Y., Colaco, T., Kashino, Z., Nejat, G., Benhabib, B.: mROBerTO: A modular millirobot for swarm-behavior studies. In: IEEE/RSJ International Conference on Intelligent Robots and Systems. pp. 2109–2114. IEEE (2016)
11. Kim, J.Y., Kashino, Z., Colaco, T., Nejat, G., Benhabib, B.: Design and implementation of a millirobot for swarm studies – mROBerTO. Robotica **36**(11), 1591–1612 (2018)
12. Kim, J.Y., et al.: A high-performance millirobot for swarm-behaviour studies: Swarm-topology estimation. Int. J. Adv. Robot. Syst. **16**(6), 1–18 (2019)
13. Drisdelle, R., Kashino, Z., Nejat, G., Benhabib, B.: Motion control of a wheeled millirobot. In: International Conference of Control, Dynamic Systems, and Robotics. (2017)
14. Eshaghi, K., Kashino, Z., Yoon, H.J., Nejat, G., Benhabib, B.: An inchworm-inspired motion strategy for robotic swarms. Robotica **39**(12), 2283–2305 (2021)
15. Chen, Z., Emami, M.R., Chen, W.: Connectivity preservation and obstacle avoidance in small multi-spacecraft formation with distributed adaptive tracking control. J. Intell. Robot. Syst. **101**, 16 (2021)
16. Yang, P., Freeman, R.A., Gordon, G.J., Lynch, K.M., Srinivasa, S.S., Sukthankar, R.: Decentralized estimation and control of graph connectivity for mobile sensor networks. Automatica **46**(2), 390–396 (2010)
17. Ji, M., Egerstedt, M.: Distributed coordination control of multiagent systems while preserving connectedness. IEEE Trans. Robot. **23**(4), 693–703 (2007)
18. Luo, W., Yi, S., Sycara, K.: Behavior mixing with minimum global and subgroup connectivity maintenance for large-scale multi-robot systems. In: IEEE International Conference on Robotics and Automation. pp. 9845–9851. IEEE (2020)
19. Giordano, P.R., Franchi, A., Secchi, C., Bülthoff, H.H.: A passivity-based decentralized strategy for generalized connectivity maintenance. Int. J. Robot. Res. **32**(3), 299–323 (2013)
20. Sabattini, L., Chopra, N., Secchi, C.: Decentralized connectivity maintenance for cooperative control of mobile robotic systems. Int. J. Robot. Res. **32**(12), 1411–1423 (2013)
21. Shetty, A., Hussain, T., Gao, G.: Decentralized connectivity maintenance for multi-robot systems under motion and sensing uncertainties. J. Inst. Navig. **70**(1), 1–20 (2023)
22. Schuresko, M., Cortés, J.: Distributed motion constraints for algebraic connectivity of robotic networks. J. Intell. Robot. Syst. **56**, 99–126 (2009)
23. Fink, J., Ribeiro, A., Kumar, V.: Robust control for mobility and wireless communication in cyber–physical systems with application to robot teams. Proc. IEEE **100**(1), 164–178 (2012)
24. Ghedini, C., Ribeiro, C.C.H., Sabattini, L.: A decentralized control strategy for resilient connectivity maintenance in multi-robot systems subject to failures. In: Springer Proceedings in Advanced Robotics 6. pp. 89–102. (2018)
25. Eshaghi, K., Rogers, A., Nejat, G., Benhabib, B.: Closed-loop motion control of robotic swarms – A tether-based strategy. IEEE Trans. Robot. **38**(6), 3564–3581 (2022)
26. Eshaghi, K., Nejat, G., Benhabib, B.: A concurrent mission-planning methodology for robotic swarms using collaborative motion-control strategies. J. Intell. Robot. Syst. **108**(2), 15 (2023)
27. Defoort, M., Veluvolu, K.C.: A motion planning framework with connectivity management for multiple cooperative robots. J. Intell. Robot. Syst. **75**, 343–357 (2014)

28. Meng, Y., Nickerson, J., Gan, J.: Hierarchical multi-robot coordination - Aggregation strategies using hybrid communication. In: Proceedings of the International Conference on Informatics in Control, Automation, and Robotics. pp. 289–295. (2006)
29. Meng, Y., Nickerson, J.V., Gan, J.: Multi-robot aggregation strategies with limited communication. In: IEEE/RSJ International Conference on Intelligent Robots and Systems. pp. 2691–2696. IEEE (2006)
30. Derbakova, A., Correll, N., Rus, D.: Decentralized self-repair to maintain connectivity and coverage in networked multi-robot systems. In: IEEE International Conference on Robotics and Automa-Tion. pp. 3863–3868. IEEE (2011)
31. Kunwar, F., Wong, F., Mrad, R.B., Benhabib, B.: Guidance-based on-line robot motion planning for the interception of mobile targets in dynamic environments. J. Intell. Robot. Syst. **47**(4), 341–360 (2006)
32. Agah, F., Mehrandezh, M., Fenton, R.G., Benhabib, B.: On-line robotic interception planning using a rendezvous-guidance technique. J. Intell. Robot. Syst. **40**(1), 23–44 (2004)
33. Borg, J.M., Mehrandezh, M., Fenton, R.G., Benhabib, B.: Navigation-guidance-based robotic interception of moving objects in industrial settings. J. Intell. Robot. Syst. **33**(1), 1–23 (2002)
34. Mehrandezh, M., Sela, M.N., Fenton, R.G., Benhabib, B.: Robotic interception of moving objects using ideal proportional navigation guidance technique. Robot. Auton. Syst. **28**(4), 295–310 (1999)
35. Engin, K., Isler, V.: Minimizing movement to establish the connectivity of randomly deployed robots. Proc. Int. Conf. Autom. Plan. Sched. **28**, 451–458 (2018)
36. Varadharajan, V.S., St-Onge, D., Adams, B., Beltrame, G.: Swarm relays: Distributed self-healing ground-and-air connectivity chains. IEEE Robot. Autom. Lett. **5**(4), 5347–5354 (2020)
37. Marchukov, Y., Montano, L.: Multi-robot coordination for connectivity recovery after unpredictable environment changes. IFAC-Pap. **52**(8), 446–451 (2019)
38. Dutta, R., Kandath, H., Jayavelu, S., Xiaoli, L., Sundaram, S., Pack, D.: A decentralized learning strategy to restore connectivity during multi-agent formation control. Neurocomputing **520**, 33–45 (2023)
39. Wagner A, Arkin R.: Multi-robot communication-sensitive reconnaissance. In: Proceedings of the IEEE International Conference on Robotics and Automation. pp. 4674–4681. IEEE (2004)
40. Liu, H., Chu, X., Leung, Y.-W., Du, R.: Simple movement control algorithm for bi-connectivity in robotic sensor networks. IEEE J. Sel. Areas Commun. **28**(7), 994–1005 (2010)
41. Roy, N., Dudek, G.: Collaborative robot exploration and rendezvous: Algorithms, performance bounds and observations. Auton. Robots **11**(2), 117–136 (2001)
42. Ulam, P., Arkin, R.C.: When good communication go bad: Communications recovery for multi-robot teams. In: Proceedings of the IEEE International Conference on Robotics and Automation. pp. 3727–3734. IEEE (2004)
43. Chaimowicz, L., Cowley, A., Gomez-Ibanez, D., Grocholsky, B., Hsieh, M.A., Hsu, H., Keller, J.F., Kumar, V., Swaminathan, R., Taylor, C.J.: Deploying air-ground multi-robot teams in urban environments. In: Multi-Robot Systems - From Swarms to Intelligent Automata. pp. 223–234. (2005)
44. Hansen, E., A., Nichele, S., Yazidi, A., Haugerud, H., Mofrad, A.A., Alcocer, A.: Achieving connectivity between wide areas through self-organising robot swarms using embodied evolution. In: IEEE Symposium Series on Computational Intelligence. pp. 875–883. (2018)
45. Baroudi, U., Aldarwbi, M., Younis, M.: Energy-aware connectivity restoration mechanism for cyber-physical systems of networked sensors and robots. IEEE Syst. J. **14**(3), 3093–3104 (2020)
46 Lee, S., Younis, M., Lee, M.: Connectivity restoration in a partitioned wireless sensor network with assured fault tolerance". Ad Hoc Netw. **24**(A), 1–19 (2015)
47. Basu, P., Redi, J.: Movement control algorithms for realization of fault-tolerant ad hoc robot networks. IEEE Netw. **18**(4), 36–44 (2004)
48. Abbasi, A., Younis, M., Akkaya, K.: Movement-assisted connectivity restoration in wireless sensor and actor networks. IEEE Trans. Parallel Distrib. Syst. **20**(9), 1366–1379 (2009)
49. Haider, N., Imran, M., Saad, N.M.: CARE: Coverage-aware connectivity restoration algorithm for mobile actor/robot networks. In: Asia-Pacific Conference on Communications. pp. 439–444. (2013)
50. Mi, Z., Yang, Y., Yang, J.Y.: Restoring connectivity of mobile robotic sensor networks while avoiding obstacles. IEEE Sens. J. **15**(8), 4640–4650 (2015)
51. Abbasi, A., Younis, M.F., Baroudi, U.A.: A least–movement topology repair algorithm for partitioned wireless sensor–actor networks. Int. J. Sens. Netw. **11**(4), 250–262 (2012)
52. Mi, Z., Yang, Y., Liu, G.: HERO: A hybrid connectivity restoration framework for mobile multi-agent networks. In: IEEE International Conference on Robotics and Automation. pp. 1702–1707. IEEE (2011)
53. Bertuccelli, L.F., How, J.P.: UAV search for dynamic targets with uncertain motion models. In: Proceedings of the IEEE Conference on Decision and Control. pp. 5941–5946. IEEE (2006)
54. Luo, X.: A regional necessity based multi-agent target search strategy for post-earthquake rescue. In: Chinese Control Conference. pp. 4903–4908. (2022)
55. Meghjani, M., Manjanna, S., Dudek, G.: Multi-target rendezvous search. In: IEEE/RSJ. International Conference on Intelligent Robots and Systems. pp. 2596–2603. IEEE (2016)
56. Meghjani, M., Manjanna, S., Dudek, G.: Multi-target search strategies. In: IEEE International Symposium on Safety, Security, and Rescue Robotics. pp. 328–333. IEEE (2016)
57. Wang, P., Meghjani, M.: Lost at Sea: Multi-searcher multi-target search. In: Global Oceans 2020: Singapore – U.S. Gulf Coast. pp. 1–8. IEEE (2020)
58. Wang, P., Meghjani, M., Chen, G.: Marine trash collection: A multi-agent, multi-target search. In: OCEANS 2022, Hampton Roads. pp. 1–7. IEEE (2022)
59. Macwan A, Nejat G, Benhabib B.: Optimal deployment of robotic teams for autonomous wilderness search and rescue. In: IEEE/RSJ International Conference on Intelligent Robots and Systems. pp. 4544–4549. IEEE (2011)
60. Vilela, J., Kashino, Z., Ly, R., Nejat, G., Benhabib, B.: A dynamic approach to sensor network deployment for mobile-target detection in unstructured, expanding search areas. IEEE Sens. J. **16**(11), 4405–4417 (2016)
61. Kashino, Z., Nejat, G., Benhabib, B.: Aerial wilderness search and rescue with ground support. J. Intell. Robot. Syst. **99**(1), 147–163 (2020)
62. Woiceshyn, K., Kashino, Z., Nejat, G., Benhabib, B.: Vehicle routing for resource management in time-phased deployment of sensor networks. IEEE Trans. Autom. Sci. Eng. **16**(2), 716–728 (2019)
63. Sheridan, P.K., Gluck, E., Guan, Q., Pickles, T., Balcıoglu, B., Benhabib, B.: The dynamic nearest neighbor policy for the multi-vehicle pick-up and delivery problem. Transp. Res. Part Policy Pract. **49**, 178–194 (2013)
64. Kashino, Z., Nejat, G., Benhabib, B.: Multi-UAV based autonomous wilderness search and rescue using target iso-probability curves. In: International Conference on Unmanned Aircraft Systems. pp. 636–643. (2019)
65. Ku S.Y., Nejat, G., Benhabib, B.: Wilderness search for lost persons using a multimodal aerial-terrestrial robot team. Robotics **11**(3), 64 (2022)
66. Kashino, Z., Kim, J.Y., Nejat, G., Benhabib, B.: Spatiotemporal adaptive optimization of a static-sensor network via a non-parametric estimation of target location likelihood. IEEE Sens. J. **17**(5), 1479–1492 (2017)
67. Katoch, S., Chauhan, S.S., Kumar, V.: A review on genetic algorithm: Past, present, and future. Multimed. Tools Appl. **80**(5), 8091–8126 (2021)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Kasra Eshaghi received the B.A.Sc. and Ph.D. degrees in mechanical engineering from the University of Toronto, Toronto, ON, Canada. His research interests include software and hardware design for robotic swarms.

Naeimeh Najafizadeh Sari received B.Sc. and M.Sc. degrees in Aerospace Engineering from Amirkabir University of Technology (Tehran Polytechnic) and the University of Tehran, respectively. She earned a second M.Sc. in Mechanical Engineering from the University of Manitoba. Currently, as a Research Assistant at the University of Toronto, her work focuses on the navigation of robotic swarms.

Cameron Haigh received the B.A.SC degree in engineering science and is completing a M.A.Sc in mechanical engineering from the University of Toronto, Toronto, ON, Canada. His research interests include trajectory planning and multi-robot systems.

Darie Roman received the B.A.Sc and M.Eng degrees in mechanical engineering from the University of Toronto, Toronto, ON, Canada. His research interests include the implementation of state estimation and motion planning algorithms in embedded systems software.

Goldie Nejat received the B.A.Sc. and Ph.D. degrees in mechanical engineering from the University of Toronto, Toronto, ON, Canada (UofT). She is a Professor with the Department of Mechanical and Industrial Engineering, UofT, and the Founder and Director of the ASBLab, UofT. She is also a Fellow with the Canadian Institute for Advanced Research. Her research interests include intelligent assistive/service robots, human-robot interactions, and semiautonomous/autonomous control.

Beno Benhabib received the B.Sc., M.Sc., and Ph.D. degrees in mechanical engineering in 1980, 1982, and 1985, respectively. Since 1986, he has been a Professor with the Department of Mechanical and Industrial Engineering, University of Toronto, Toronto, ON, Canada. His current research interests include the design and control of intelligent autonomous Systems.