Algorithms for coordination of autonomous underwater drones searching for a hidden object when no long-range communication is allowed

D. N. Fedyanin
Researcher, V.A.Trapeznikov Institute of Control Sciences, Moscow, Russia
E-mail: dfedyanin@inbox.ru

Abstract. The paper is devoted to comparison of the efficiency of information exchange by autonomous mobile agents when searching in a limited coverage area. The special feature of the problem is to consider the case of operation within the prohibition of usage of radio, light, and acoustic communication channels at a great distance, as well as without surfacing. The paper compares the following algorithms for the work of groups of autonomous robots, based on a preliminary division of area into sectors and survey of robots in their sectors using the “breadth-first search” method.

1. Introduction
The paper compares the following algorithms for the work of groups of autonomous robots, based on a preliminary division of area into sectors and survey of robots in their sectors using the “breadth-first search” method. In the proposed four algorithms for information exchange when solving the search problem, their strengths and weaknesses are studied. A computer model was built for simulation, and a preliminary analysis was performed. The paper is devoted to comparison of the efficiency of information exchange by autonomous mobile agents when searching in a limited coverage area. This problem arises when underwater vehicles search for a source of pollution on the surface of a flat bottom.

The task of the group of robots is to find the location of the source of pollution, report this to the base and return all robots to the base. We will refer the source of pollution as the object assuming universality. The special feature of the problem is to consider the case of operation within the prohibition of usage of radio, light, and acoustic communication channels at a great distance, as well as without surfacing. The permitted communication channel is data transmission within the distance of the size of the robot itself [2].

The task consists of several interconnected subtasks:
1. Find the object [1]
2. Supply fuel for this [4]
3. Inform all robots when object is found [6]
4. Stay hidden [5]

It is important to solve them at once but, for the sake of simplicity, sometimes we use restriction which reduces the complex tasks to solution of only single task. We discuss it after the description of the algorithms. Although the connection between the 1st and the 2nd subtasks is most obvious, we will
pay almost no attention to it; instead, we will focus on the connection between the 3rd and 4th subtasks. We created a software application in order to partially simulate the algorithms, see Fig 1.

![Screenshot of an interface of the software.](image)

**Figure : 1: Screenshot of an interface of the software.**

2. **Definitions and formal problem**

**Environment properties**
- Location of the object is $z$,
- We assume that the base is located at point (0; 0) without dramatical loss of generality, because all five algorithms might be adapted to 3D case,
- $dx$ — the maximum error in position designation.

**Drone types**
- $S(x)$ is an observable area for the drone at point $t$ if the drone is at point $x$
- $R(x)$ is the area where data transmission is available if the drone is at point $x$
- $p$ is the probability of a drone failure
- $c_m$ is a drone fuel consumption while searching
- $c_r$ is a drone fuel consumption while moving
- $v_s$ is the absolute velocity of search of the a drone
- $v_m$ is the absolute velocity of movement a drone

**Drone informational properties**
- $e_i(t)$ is the indicator if $i$-th drone at time $t$ believes that the object is found
- $h_i(T)$ — a set of indexes of agents at a moment such for every $j$ in this set the $i$-th drone knows $e_j(t)$ for any $t<T$
- $x_i(t, h_i(t), e)$ is the location of the $i$-th drone at time $t$

**Conditions**
1. Search area condition
   $x_i(t) \in S, \forall t, i$
2. Initial condition
\[ x_i(0) = y_i(0) = e_i(0) = 0, \forall i \]

3. Convergence condition
\[ \forall i, t > T_j, x_i(t) = 0, e_i(t) = 1 \]

3. Epistemic model and phantom agent
The focus of the paper is on the information spread. We consider the following types of information change channels:

1. Information can change only after direct physical contact
   - with object itself
   - with base station
   - with the drone that found the object
   - another drone

2. Information can change only after direct physical contact or after absence of direct physical contact if that was planned
   - with the drone who found the object
   - another drone

3. Another variant
We have found that it is dramatically important to pay attention to this aspect. At first, we thought that we could consider only function \( e_i(t) = 0 \) if \( i \)-th drone believed that the object was found, and \( e_i(t) = 1 \) if otherwise.

Search success condition: there are \( t, i \) such that \( z \in S(x_i(t)) \)

Information condition

1. The condition that it is impossible to know the location of the object before it is found by at least one drone
\[ e_i(t) = 0, \forall t, j : \neg \exists t_0 < t : z \in S(x_j(t_0)) \]

2. Condition for absolute memory location of the object
\[ e_i(t) = 1, \forall t > t_i : x \in S(x_i(t)) \]

3. Condition of information transfer only by contact (only for some algorithms)
\[ e_i(t) = 1, \text{ if there exist } i_1, t_1, i_2, \ldots, i_n, t_n \text{ such that } e_{i_1}(t_1) = 1, e_{i_2}(t_2) = 1, \ldots, e_{i_n}(t_n) = 1 \]

4. If there is a sequence 1 of contacts such that the drones approach each other at an interaction distance
\[ |x_{i_k}(t_k) - x_{i_{k+1}}(t_{k+1})| < R, \]

5. The boundary condition for this constraint
\[ |x_i(t_n) - x_j(t_n)| < R. \]

We assume that \( e_i(t) = 1 \) makes \( i \)-th drone move at velocity \( v_m \) unless it is already at the base since there is no need in search. \( e_i(t) = 0 \) does not mean that \( i \)-th drone moves at velocity \( v_s \) since we can push drones to move to check if other drones are not broken or regularly return to the base.

4. Optimization criteria
Optimization criteria are in Table 1.

| Table 1. Optimization criteria |
|--------------------------------|
| Criterium | Maximization | Additional restrictions |
|-----------|--------------|------------------------|
| 1. Search time | \( \min T_1 \) | \( \exists t: \forall t > T_1, e_i(t) = 1 \) |
| 2. Time of notification after the object is found | \( \min dT - \min T_1 \) | \( \exists t: \forall t > T_1, x_i(t) = 0, e_i(t) = 1 \) \( \exists t: \forall t > T_1, e_i(t) = 1 \) |
| 3. Return time after the object is found | \( \min \max_j T_j - \min T_1 \) | \( \forall t > T_j, x_i(t) = 0, e_i(t) = 1 \) \( \exists t: \forall t > T_1, e_i(t) = 1 \) |
4. Optimizing overall fuel consumption

\[
\min_x \max_z \sum_i \int_0^{\infty} c \left| \frac{d}{dt} x_i(t) \right| dt
\]

5. Optimizing radar alert probability

\[
\min_x \max_z \sum_{ij} \int_0^{\infty} \left( x_i(t) - x_j(t) \right)^2 dt
\]

4.1. The regular return algorithm

The regular return algorithm involves the return of robots to the base synchronously to the given point in time or after successful finding of the source of pollution. The regular return algorithm is very expensive in terms of fuel and search time. The time of the last and the first drones can be both very long and short, but they increase significantly as the search moves away from the base.

![Fig. 2. Schematics of the drone motions in the regular return algorithm.](image)

4.2. The algorithm with a courier

The algorithm with a courier involves selecting a special robot moving among robots and checking if the robots have found the source. If the robots find that the special robot has not made contact at the specified time, this is a signal that the search is completed and all robots return to the base. The algorithm is fuel-consuming and leads to an increase of search time, because one drone is used only for transmitting information.

![Fig. 3. Schematics of the drone motions in the algorithm with a courier.](image)
4.3. The high-speed cascade algorithm.
A robot that has successfully found the source moves to the nearest robot along the shortest path, then the robots moving along special trajectories inform the other robots in a cascade, while one of the robots moves the shortest way to the base. Algorithm (3) surpasses the other three in time it takes all drones to return to the base, but is inferior or equal (if modified) in the return time of the first drone, and in terms of fuel consumption it loses significantly to the algorithm from the rendezvous.

Fig. 4. Schematics of the drone motions in the high-speed cascade algorithm.

4.4. The rendezvous algorithm
Robots synchronize and exchange information when they meet, if a source is found, the robot moves to the base, and the other robots, if they have not found a neighboring robot at a given time (the movement of robots will be calculated if the trajectories are known), go to the base. It is also a cascade algorithm, but unlike Algorithm 3, the propagation speed is slower (for example, when the territory is large and regular meetings of robots are rare). During the analysis, it was shown that algorithms 3 and 4 have an advantage, with the advantage of algorithm 3 in fuel economy, and 4 - in a shorter task execution time. The algorithm wins in fuel economy and the first drone criterion, but loses in the last drone criterion. The rendezvous algorithm provides a signal propagation speed almost equal to the speed of the drones themselves and does not require physical transmission of a signal about the location of pollution.

Fig. 5. Schematics of the drone motions in Algorithm 4.

4.5. The “mob” algorithm
All drones move close to each other and search in a way like a single drone would use. The algorithm dominates or at least equals at fuel consumption, but dramatically fails at radar alert probability criterion.
5. Noise proof modifications

The algorithms except for the first one are very sensitive to the noise of location determination. The regular return algorithm is noise-proof if noise is small enough to allow drone to return to the base. If $S$ is smaller than a circle with radius $R$.

In this case, the solution for the algorithms is introducing waiting time or time for looking for another drone scanning the area. The modification for the high-speed cascade is scanning around the point where another drone is supposed to be. This idea is well-known. This modification works fine for the rendezvous algorithm. Unfortunately it is applicable only if $\nu_m >> \nu_s$. If it is not, then the modification for the rendezvous algorithm is assigning a role to the drones at the rendezvous point. A drone is just waiting and another drone is trying to find him. The rendezvous fails if drones do not find each other during some fixed waiting time. The larger the noise is, the larger the waiting time should be.

The modifications for the high-speed cascade are more complicated. There are several options depending on possible noise. A drone could try to follow him using $\nu_m > \nu_s$ or just stop and wait until the drone returns to this area. This version almost satisfies the requirement but there is no guarantee they will meet. The universal variant for drones is to stop at some points at fixed times so the drone which is looking for them could come to such points and look around as for the rendezvous algorithms. The difference is that the drone which waits does not activate if he is not found by a drone at such points. Another difference is that there should be more points for such stops.

The mob algorithm might be rather noise-proof since drones are near each other during the search and can be easily solved when $\nu_m >> \nu_s$. The problem occurs when two velocities are very close. In this case there are two strategies for the high-speed cascade: try to follow drones or stay nearby and wait for their return.

We assume that search algorithms have a mixing property – it is always possible to find two points at a searching drone trajectory which are closer than $3RS$. It is true for many quasi-periodic search algorithms.

6. Failure proof modifications

There are several ways to find out that a drone is broken, but only one is reasonable.

- The first one is making drones come back to the bases to inform that they are not broken. The problem is too much fuel and time consumption.
- The second is to send drones from time to time to touch each other and it requires a plan to cover all the situations. The problem is too much fuel and time consumption. It is less than the first way but less stable for noise too.
- Duplicate all functions by sending two drones. It is safe but decreases all other parameters of the algorithm,
- Rendezvous version of the second way. If a drone finds out that its partner has not come to the rendezvous, there are two variants
  a. come to base if there is no partner; then it means that its partner is broken and the drones come back to the rendezvous point.
  b. follow the way of its partner to check if it is broken at a point at its trajectory.
All versions will be studied in future works. As we have broken the drones, we will need to take them back by drones and it is slightly a separate problem. The interesting influence of such need is required adaptation of algorithms when the number of drones is decreasing. The algorithms look tentative to this need except for Algorithms 1 and 5 which are able to handle it easily.

7. Cluster versions of algorithms
All of them use the idea of dividing the set of drones into several groups and maybe further into subgroups.
1. There might be a meeting point not on the base but in the center of the set of all drones.
2. There might be several meeting points if the drones are divided into several groups.
3. Once the drones of one group find an object, a cascade algorithm starts.
4. Since there are already meeting points, drones could come to them instead of reaching all drones individually.

Fig. 7. Ideas of the cluster version. Solid lines are trajectories of the centers of the clusters, dotted lines are trajectories of drones in a cluster and between clusters.

References
[1] Chkhartishvili A.G., Shikin E.V. Dynamic object search. Geometric view of the problem. Funkcional’naia i prikladnaya matematika, 1:4 (1995), pp. 827–862 (in Russian)
[2] Leung K. Y. K., Barfoot T. D., Liu H. H. T. Decentralized localization of sparsely-communicating robot networks: A centralized-equivalent approach, IEEE Trans. on Robotics. 2009, vol. 26, no. 1, pp. 62-77.
[3] Ren W., Atkins E. Distributed multi-vehicle coordinated control via local information exchange, Int. J. of Robust and Nonlinear Control: IFAC Affiliated Journal, 2007, vol. 17, nos. 10-11, pp. 1002-1033.
[4] Pfingsthorn M., Birk A., Bülow H. An efficient strategy for data exchange in multi-robot mapping under underwater communication constraints, 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems IEEE, 2010, pp. 4886-4893.
[5] Couceiro M. S. et al. Darwinian robotic swarms for exploration with minimal communication, 2013 IEEE Congress on Evolutionary Computation, IEEE, 2013, pp. 127-134.
[6] De Hoog J., Cameron S., Visser A. Autonomous multi-robot exploration in communication-limited environments, Proceedings of the Conference on Towards Autonomous Robotic Systems, 2010, pp. 68-75.