Measurement 45 (2012) 971–987

Contents lists available at SciVerse ScienceDirect

Measurement

journal homepage: www.elsevier.com/locate/measurement

Autonomous multiple teams establishment for mobile sensor networks by SVMs within a potential field

Sedat Nazlibilek *

Atilim University, Engineering Faculty, Department of Mechatronics, Ankara, Turkey
Bilkent University, NANO Technology Research Center, 06800 Ankara, Turkey

ARTICLE INFO

Article history:
Received 23 June 2011
Received in revised form 13 October 2011
Accepted 31 January 2012
Available online 15 February 2012

Keywords:
Algorithms
Measurement
Networks
Magnetic field measurement
Robots
Vectors

ABSTRACT

In this work, a new method and algorithm for autonomous teams establishment with mobile sensor network units by SVMs based on task allocations within a potential field is proposed. The sensor network deployed into the environment using the algorithm is composed of robot units with sensing capability of magnetic anomaly of the earth. A new algorithm is developed for task assignment. It is based on the optimization of weights between robots and tasks. The weights are composed of skill ratings of the robots and priorities of the tasks. Multiple teams of mobile units are established in a local area based on these mission vectors. A mission vector is the genetic and gained background information of the mobile units. The genetic background is the inherent structure of their knowledge base in a vector form but it can be dynamically updated with the information gained later on by experience. The mission is performed in a magnetic anomaly environment. The initial values of the mission vectors are loaded by the task assignment algorithm. The mission vectors are updated at the beginning of each sampling period of the motion. Then the teams of robots are created by the support vector machines. A linear optimal hyperplane is calculated by the use of SVM algorithm during training period. Then the robots are classified as teams by use of SVM mechanism embedded in the robots. The support vector machines are implemented in the robots by ordinary op-amps and basic logical gates. Team establishment is tested by simulations and a practical test-bed. Both simulations and the actual operation of the system prove that the system functions satisfactorily.

1. Introduction

In this paper, a new method and algorithm for team establishments within a mobile sensor network (SN) is proposed. The sensor network deployed into the environment using the algorithm is composed of robot units with sensing capability of magnetic anomaly of the earth. They utilize KMZ51 un-isotropic magneto resistive sensors. The sensors are combined to obtain a convolution mask for steepest descent in the magnetic anomaly region. The innovation in this method is to establish more than one team based on tasks available in the region of operation. A task means that it is a magnetic anomaly of earth's magnetic field caused by some dangerous mines. The aim is to detect these mines. There may be several magnetic anomalies within a region. They have to be classified in an intelligent way. This can be achieved in several ways. Why we prefer a new method to identify these sub-regions by multiple robotic teams is that in the previous study [1] it was observed that some of the mobile units had stuck around some sub-regions. For example, in Fig. 1, the three yellow[1] robots swarm into the sub-region where an AT mine is buried, while the other two blue robots approach into another

* Address: Bilkent University, NANO Technology Research Center, 06800 Ankara, Turkey. Tel.: +90 312 290 3050; fax: +90 312 290 1015.
E-mail address: nazlibileksedat@yahoo.com

[1] For interpretation of color in 1–9 and 14–16, the reader is referred to the web version of this article.

0263-2241/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
doi:10.1016/j.measurement.2012.01.046

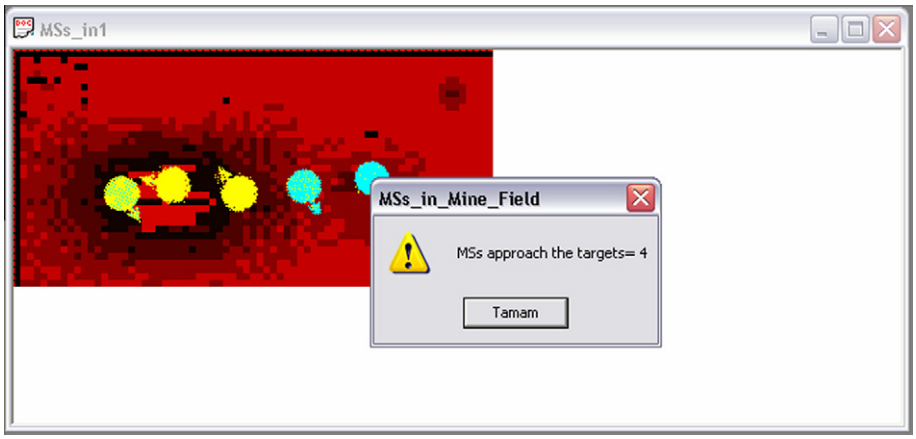


Fig. 1. Groups of robotic units approaching two separate objects.

sub-region where an AP mine is buried. They constituted small groups of robots based on the information gathered by the magneto-resistive sensors mounted on them and calculating the gradient of steepest decent in the magnetic anomalies. There was no external intervention to the motions of the robots. They performed their behavior autonomously. The behavior of the mobile sensor units are synchronized in time by a time division multiple access (TDMA) method [1]. Some of these groups detected important objects but the others could not. The robots within the mobile SN had been separated in an inappropriate manner. This caused a waste of resources in most of the cases. It motivated us to create teams of robotic units in an appropriate way based on the importance of sub-regions. In this paper, we modeled the requirement as a task assignment problem.

In recent years, mobile sensor networks are used in a wide variety of applications such as establishing formations, imitating the behavior of some animals, detecting objects, performing search and rescue activities, area coverage, surveillance and reconnaissance, or controlling streetlights for energy savings. In this study we try to achieve autonomously creation of more than one robot teams allocated into several tasks available within an area of operations where more than one anti-tank (AT) and/or anti-personnel (AP) mines exit. The work is motivated by the detection of these buried mines (anti-personnel and anti-tank) at the border regions for clearing purposes. The mines buried across these regions are hard to find because of the fact that there are no plans available any more or the mines had changed their locations as a result of some geological, natural and/or manmade effects. In our application, an area of several square meters, normally 1.5 m $\times$ 1.5 m, is scanned by magneto-resistive sensors and the buried objects are detected by the anomaly of the measured earth magnetic field.

Within the scanned region, there may be more than one buried objects creating magnetic anomalies. In our previous study, we classified the objects one by one based on the data collected by mobile sensors acting over the region [1]. It was an effective approach for mine detection, but as we observed during operation of the mobile sensor network that some of the robots were grouped around some other buried objects if available within the same region. This is a condition that can frequently been encountered in real applications, therefore we think that we can approach the problem in a different way.

The sensor network deployed into the environment using the algorithm is composed of robot units with sensing capability of magnetic anomaly of the earth. They utilize KMZ51 un-isotropic magneto resistive sensors. The sensors are arranged in a 3 $\times$ 3 sensor matrix to implement a convolution mask for steepest descent in the magnetic anomaly region. The mask is used to determine the gradient of the field as a hardware element. The robots perform rotation and translation motion at the end of each operation period based on the direction of the gradient vector. The mobile units are synchronized by using the method called time division multiple access (TDMA). In this method, a time slot is assigned to each robotic unit to allow them to make movements within its time slot.

We can think of the problem as a task allocation problem which has received significant interest in recent years. As seen in Fig. 2, the objects found in the region can be considered as tasks and the aim will be to assign multiple robots to these tasks based on an appropriate technique.

There are two common methods applied for task allocation, namely, behavior-based [2–5] and market-based approaches [6,7]. One of the earliest behavior-based method is the so called the ALLIANCE Efficiency Problem (AEP) which is an NP-Hard problem [3]. The other famous behavior-based architecture is Broadcast Local Eligibility (BLE) presented in [4]. In general, behavior-based method is a control methodology in which mobile agents are controlled through the principled integration of a set of interacting behaviors in order to achieve desired system-level behavior. Behavior-based approaches are an extension of reactive architectures and also fall between purely reactive and planar-based extremes [5].

The market-based (or frequently called auction-based) approach is another well known method for solving task allocation problem. The famous examples of marked-based methods are the M + system in [6] and the MURDOCH in [7]. These methods are based on or a variant of the well-

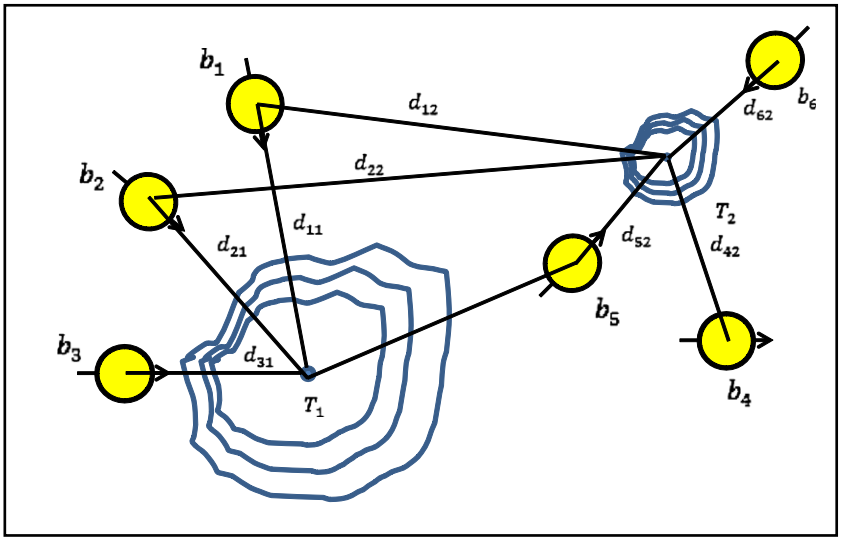


Fig. 2. Basic approach for task assignment to multiple robotic teams.

known Contract Net Protocol (CNP) [8]. A survey on market-based approaches can be found in [9], Complex task allocation problems are dealt with in [10] and [11]. In general, the market-based task allocation approach is based on the principles of a market economy that can be applied to multi-robot coordination where each robot acts to maximize its individual profit and simultaneously improves the efficiency of the team. In this approach, two roles are played dynamically by robots: auctineers and bidders [9,10,12]. The auctioneer is the robot or central station in charge of announcing the tasks and selecting best bid from the bidders. The best bid is the one with lowest cost. The cost is equal to the distance from the robot to the task. The bid is a quantity that reflects how much it will cost the robot to go to a certain waypoint, such as the Euclidean distance or the traversability index. The properties of the market-based task allocation can be listed as follows: As teams of robots as participants in a virtual economy; robots are contracted to complete required tasks in exchange for payment; each robot has well-defined cost and revenue functions to compute the expected gains and losses for performing tasks; they work to maximize their individual profits; in a market, trading tasks via auctions; auctions call for bids; the robots that can perform the tasks for the best price are awarded the resulting contracts.

Another popular method for task allocation is the use of vacancy chains [13,14]. The vacancy chains (VCs) are a resource distribution strategy common in human and animal societies. Vacancy chains algorithm is uses local task selection, reinforcement learning for estimation of task utility and reward structures based on the vacancy chain framework. Three requirements are listed for resource distribution through VCs: the resource must be used by only one individual; a vacancy is required before an individual takes a new resource unit; and vacancy resource units must be scarce and many individuals must occupy sub-optimal unit. In [13], it is demonstrated that VCs could be used to optimize the performance of a group of robots when the given task conforms to the above three requirements. In fact, VCs often disappear when information is widely available, because information is likely to attract applicants who engage in a competition [14]. A vacancy may be caused by a retirement, which triggers a chain of vacancies through subordinates. At the beginning only one subordinate is available. However, when two or more equally qualified persons apply for the same position, the resource is allocated by the labor market. The fact that a VC is identified does not imply that it is the prime move of resource allocation. This distinction has never been explicitly stated in empirical research on VCs. All empirical investigations have focused on situations where VCs, at least in their initial steps, did operate as allocation mechanisms. Scarce resource is allocated by means of VCs.

In [2], it is claimed that the multi-robot task allocation could be reduced to an instance of the Optimal Assignment Problem (OAP) [15] that can be casted as a linear optimization problem.

A well-known method for task assignment called the "Hungarian Method" is given in [16]. It can solve the OAP faster. It develops a computational method that uses the dual linear program in a particularly effective manner.

The simple task assignment problem stated in [16] is as follows: n individuals (denoted by $i = 1, 2, \ldots, n$) are available for n jobs (denoted by $j = 1, 2, \ldots, n$). They qualify for jobs represented effectively by a (nxn) qualification matrix **Q** in which the rows stand for individuals and columns for jobs and the entries $q_{ij} = (1 \text{ or } 0)$ representing ratings indicating that a worker is qualified or not respectively. Then the simple assignment problem asks: What is the largest number of 1's that can be chosen from **Q** with no two chosen from the same row or column?

The general assignment problem is as follows: Suppose n individuals $(i = 1, 2, \ldots, n)$ are available for n jobs $(j = 1, 2, \ldots, n)$ and that a rating matrix $\mathbf{R} = (r_{ij})$ is given, where r_{ij} are positive integers, for all i and j. An assignment consists of the choice of one job j_i for each individual i such

that no job is assigned to two different men. The General Assignment Problem asks: For which assignments is the sum of the ratings $(r_{1j_1} + r_{2j_2} + \cdots + r_{nj_n})$ largest?

The Hungarian method and the other task assignment methods mentioned above are generally dealing with the assignment of a single individual to a single job.

We have two kinds of requirements, namely we can either assign single robot to a single task or multiple robots to a single task. In most of the cases, the latter requirement is more frequent. In our application, we want to achieve a collective behavior for detecting any object found in the searched region. We want multiple robots to be swarmed autonomously into an area where a magnetic anomaly created by a ferromagnetic object is available. Every anomaly can be considered as an area where a mission will be performed. Therefore, an anomaly can be considered as a task to be assigned to multiple robots. Since there may be more than one object (i.e., task) in the region, we need to have more than one robot team in the region of operation. Therefore, we need to create a couple of robots teams based on the number of tasks available in that region. We have to classify the teams of robots as separate teams having specific tasks to be performed. That means that we need to solve a task assignment problem to multiple teams. Each team will have a specific task. The teams will be established depending on the priorities of the tasks and the skill ratings of the robots. That is, each robot r_i, $i = 1, \ldots, n$ will have a skill ratings $s_i,\ i = 1, \ldots, n$ and each task $t_j,\ j = 1, \ldots, m$ will have a priority, $p_j,\ j = 1, \ldots, m$.

Every robot in the mobile SN is initially assigned to a task. We developed a new algorithm for initial assignment of the tasks. After initial assignment, the teams are established by use of support vector machine (SVM) technique. This will speed up the establishment of teams within the mobile SN. Also, it will help the robots to update their tasks periodically during the course of the actions. The duration of the period can be determined a priori. This kind of updating action may improve the overall performance of the system in terms of performance criteria determined in the algorithm. We call this algorithm as the "mod-median" algorithm. Since it sorts out the unworthy robots the related task based on mod and median of the performance values of the robots, and it assigns robots deserving this task.

The structure of this paper is as follows: Section 2 gives the problem definition in detail and the algorithm development for task assignment. In Section 3, methods of team establishment are explained. The mission vectors are created by two methods, namely, the Skill rating – Priority (SP) Method and the Most Skillful Robot (MSR) Method. The theory and implementation of support vector machines are also given in this section. Section 4 gives the experimental results. The conclusion is given in Section 5.

2. Problem definition and algorithm development

2.1. Problem definition

The situation is that there are n robots and m tasks within a mission region. The robots have some skills expressed as skill ratings and the tasks have some priorities. As it is stated, the aim is to assign groups of robots to appropriate tasks such that the overall performance becomes optimum. To do this, an algorithm called the "mod-median" algorithm is developed. The algorithm has two versions. The first one makes an assignment in such a way that a single robot is assigned in a single task. No other robots are assigned to the same task. The second version makes an assignment such that multiple robots (deserved ones) can be assigned to a single task in optimum way.

The robot task assignment problem can be stated as follows: Given robots with skill ratings $(r_i, s_i),\ i = 1, \ldots, n$ and tasks with priorities $(t_j, p_j),\ j = 1, \ldots, m$, assign robots to tasks so as to maximize the overall expected performance. Normally, the expected performance is the weighed sum of the utilities, which is a concept borrowed from economics, that is the internal estimation of the value or the cost of execution of an action by an individual, and priorities belonging to the tasks.

The course of action by robot teams is divided into time periods, T. One period (T) is also divided into two parts as initial assignment (T_i) and action (T_a), where $T = T_i + T_a$. In the initial assignment sub period, the task assignments are achieved by the so called "mod-median" algorithm to be developed here. In the action sub period, (T_a) the robots will be grouped into teams by means of SVMs. At the end of the time period T, a new period with the same structure begins again. The action period is chosen long enough such that robots in a team can do several rotational and translational motions within the region of anomaly. The assignments can be updated again at the beginning of the new period. The duration for T can be a time interval passing at least between the start of action and the first change in rotation angle of any robot within the team. Normally, it is determined by the user based on previous experiences.

The map of r_i's to t_j's is shown in Fig. 3. In this mapping, the weights are the sum of skills of robots s_i's and the priorities p_j's, that is, $w_{ij} = s_i + p_j$, $i = 1, \ldots, n$ and $j = 1, \ldots, m$. That is, both the skills and the priorities of the tasks must be high. The performance in this problem is the sum of the weights, $\sum_{ij} w_{ij}$. Assign robots to tasks so as to maximize the overall expected performance (i.e., the sum of the weights), that is, max $\sum_{ij} w_{ij}$.

If we assign a single robot to a single task, and if $n > m$, then some robots will be idle, else if $n < m$, then some of the tasks will be empty. But, in any case, the performance must be greatest in single-to-single assignment. Let's define the weight matrix as (boldface capital letters represent matrix quantities and boldface small letters represent vector quantities):

$$\boldsymbol{W} = \begin{bmatrix} w_{11} & w_{12} & \ldots & w_{1m} \\ w_{21} & w_{22} & \ldots & w_{2m} \\ \ldots & \ldots & \ldots & \ldots \\ w_{n1} & w_{n2} & \ldots & w_{nm} \end{bmatrix} \tag{1}$$

In case we assign multiple robots (or all of them) to a single task, then we have to determine which mapping will give the greatest performance. That is, among the column vectors:

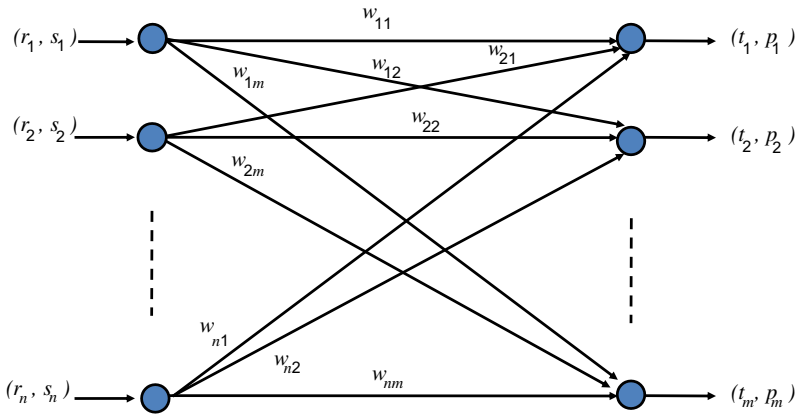


Fig. 3. The mapping of the robots to the tasks.

$$\boldsymbol{w}_1 = [w_{11}w_{21}\ldots w_{n1}]^T, \boldsymbol{w}_2 = [w_{12}w_{22}\ldots w_{n2}]^T, \ldots, \boldsymbol{w}_m = [w_{1m}w_{2m}\ldots w_{nm}]^T \quad (2)$$

We can determine which one is the greatest. Then we assign all the robots to that task. However, in this situation, we have assigned all of the robots only to one single task. This is not our aim. We try to assign only the robots that deserve mostly to that job and eliminate the others. How do we achieve such elimination? This can be done by sorting the elements of the vectors $\boldsymbol{w}_j = [w_{1j}w_{2j}\ldots w_{nj}]^T$ from the greatest to the smallest. Then, we can assign the ones above the median to the tasks and leaving the unassigned ones as idles. What will happen to the idle robots? They will wander around the region and search targets around them. In some time, they may gain a skill, for example approaching a target and deserve a job. Since we do initial task assignments at the beginning of each period T, we review the task assignments again and we start a new cycle. In this way, the task assignment becomes a dynamic assignment.

We can show this approach by an example given in the following.

2.2. Example

Let the number of robots be 3 and the number of tasks be 2. The robot-skill and task-priority pairs are $(r_1, s_1) = (1, 5)$, $(r_2, s_2) = (2, 3)$ and $(r_3, s_3) = (3, 1)$, and $(t_1, p_1) = (1, 5)$ and $(t_2, p_2) = (2, 2)$. (a) Assign single robot to single task; (b) assign multiple robots to single task. The matrix:

$$\boldsymbol{W} = [\boldsymbol{w}_1 \quad \boldsymbol{w}_2] = \begin{bmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \\ w_{31} & w_{32} \end{bmatrix} = \begin{bmatrix} (s_1+p_1) & (s_1+p_2) \\ (s_2+p_1) & (s_2+p_2) \\ (s_3+p_1) & (s_3+p_2) \end{bmatrix} = \begin{bmatrix} 10 & 7 \\ 8 & 5 \\ 6 & 3 \end{bmatrix}$$

(a) The single robot to single task assignment: The largest weight mapped to t_1 is $w_{11} = 10$, therefore the robot-task pair is (r_1, t_1). Similarly, the largest weight mapped to t_2 is $w_{12} = 7$. This weight belongs to r_1. However, r_1 is already assigned to t_1, therefore it is discarded. The next larger one is $w_{22} = 5$. Therefore, the assignment pair is (r_2, t_2). Notice that, initially it is determined which of the applications is the largest.

(b) Multiple robots to single task assignment: The elements of the vector w_1 is sorted from largest to smallest as $\boldsymbol{w}_1 = [10 \quad 8 \quad 6]^T$. The median is $w_{21} = 8$. Take $\boldsymbol{w}_{11}$ and w_{21} which are above the median (included). Assign (r_1, t_1) and (r_2, t_1). Now, sort the second column vector, $\boldsymbol{w}_1 = [7 \quad 5 \quad 3]^T$. The median is $w_{22} = 5$. But, the robot r_1 with largest weight $w_{21} = 7$ has already been assigned to t_1 and the robot r_2 in the mapping w_{22} has already been assigned to t_1. Therefore, we have to delete these assignments. Only remaining assignment is w_{32}. But, $w_{32} = 3$ is below the median. Therefore, the robot r_2 cannot be assigned to a task. Only the robots with mapping greater than the median can be assigned to an appropriate task. In this case, r_2 is idle and t_2 is empty.

2.3. Algorithm development

In this work, we develop two versions of the assignment algorithm. In the first version, a single robot to single task (SRST) assignment can be achieved.

The SRST assignment algorithm is given below:

Step 1. Create the $(m \times n)$ $\boldsymbol{W} = [\boldsymbol{w}_1 \quad \boldsymbol{w}_2 \quad \cdots \quad \boldsymbol{w}_{m-1} \quad \boldsymbol{w}_m]$ where $\boldsymbol{w}_j$'s are column vectors whose entries are the weights (that are the sum of skill s_i and priority p_j related to robot-task pair $[s_i, p_j]$ respectively) of mapping $w_{ij} = s_i + p_j$, m is the number of tasks. The dimension of the column vectors n is the number of robots.
Step 2. Sort the column vectors from the largest to smallest.
Step 3. Get the greatest among the column vectors $\boldsymbol{w}_j$.
Step 4. Find the largest element w_{ij} of the vector $\boldsymbol{w}_j$.
Step 5. Perform the mapping of i to j.

Step 6. Delete w_{ij} of the $[r_i, t_j]$ pair (i.e., discard r_i and t_j).
Step 7. If all the assignments are completed, then stop.
Step 8. Increment the index of column vector.
Step 9. Go to Step 3.

The second version of the algorithm can perform multiple robots to single tasks (MRST) making the overall performance maximum. This MRST algorithm is called "mod-median" algorithm. The steps for the mod-median MRST algorithm are given in the following.

MRST Assignment Algorithm:

Step 1. Create the $(m \times n)$ $\boldsymbol{W} = [\boldsymbol{w}_1 \quad \boldsymbol{w}_2 \quad \cdots \quad \boldsymbol{w}_{m-1} \quad \boldsymbol{w}_m]$ where w_j's are column vectors whose entries are the weights (that are the sum of skill s_i and priority p_j related to robot-task pair $[s_i, p_j]$ respectively) of mapping $w_{ij} = s_i + p_j$, m is the number of tasks. The dimension of the column vectors n is the number of robots.
Step 2. Sort the column vectors from the largest to smallest.
Step 3. Get the greatest among the column vectors $\boldsymbol{w_j}$.
Step 4. Find the median element w_{ij} of the vector $\boldsymbol{w_j}$.
Step 5. Perform the mapping of i to j.
Step 6. Delete r_i and all of the connections, w_{ij}'s of the r_i to the tasks.
Step 7. If all the assignments are completed, then stop.
Step 8. Increment the index of column vector.
Step 9. Go to Step 3.

The skill ratings and priorities must be set before the operation or during the operation. They are either set by the user before the operation or determined automatically during the course of action. For autonomous operations, the initial values of the skill ratings and priorities are set to zero.

The skill ratings are related to the distances from the robots to the tasks. The shorter the distance is, the more skillful the robot. Hence, the skill ratings can be chosen as the inverse of the distance between the *ith* robot and the *jth* task, i.e.,

$$s_{ij} = \frac{1}{d_{ij}}. \tag{3}$$

Here we use double index to show the skill ratings of any robot with a task. A robot may have different skill ratings for each individual task within a region.

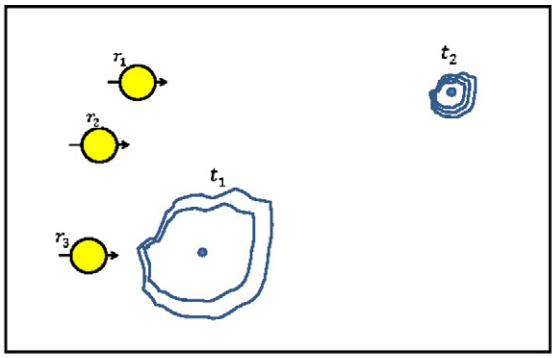
The priorities of the tasks are the values mainly determined from the number of robots aiming at the target objects. It is the measure of importance of the task. A priority can be defined as the number of robots aiming to that target multiplied by the inverse of the size of the object. That is,

$$p_j = \frac{v_j}{n} \times \frac{1}{c_j} \tag{4}$$

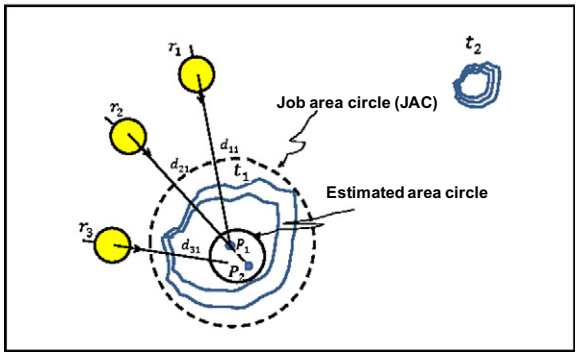
where c_j is the diameter of the jth target object and v_j is the number of robots aiming at this target. $\frac{v_j}{n}$ is the ratio of the number of robots approaching the target j to the total robots n. Since the weights are the sum of skill ratings and priorities, we take the inverse of the diameter in order to make the units consistent. In this case, both the skill ratings and the priorities are in 1/m. The units are consistent now and they can be added. However, there is a problem with this definition. Although it seem that the importance of the target will increase when a number of robots swarm in a confined region, the priority value will go to infinity when the value of the diameter goes to zero. This leads to a misleading operation in the case of, for example, when there is a small object that has a potential of giving rise to a false alarm. In order to prevent such a problem, the diameter value is fixed to a constant value. It is taken as the diameter of an AT mine in this application. Then, the priority is only depended on the number of robots desiring to arrive at the target. The definition of the priority is then:

$$p_j = \frac{v_j}{n} \times \frac{1}{c} \tag{5}$$

where c is the diameter of an AT mine. This is the so called "job area circle (JAC)". In this application, JAC diameter is chosen as the diameter of AT mine (Fig. 4).



(a) The time step is k=0 (Initial values of the skill ratings and priorities are set to zero)



(b) The time step is k=1

Fig. 4. (a) The robots are arbitrarily located within the area of operation; (b) the robots detect a magnetic anomaly and can make rotations based on the direction of gradient of the field. After the rotational motion, forward intersections are performed and the points P_1 and P_2 are obtained. Then the distances and the job area are estimated. Based on these values, skill ratings and priorities are updated.

The distances between the robots and the intersection points are determined and set as the skill ratings. They are put into the databases of the robots in the form of vectors. Also the coordinates of the intersection points are put into the database vectors. The database vector is called the "mission vector (MV)". The MV is defined clearly in Section 3. The MV is used for the creation of teams by means of SVM as explained later. The flow chart giving the overall operation is given in the Appendix.

3. Team establishment

In this work, depending on the tasks available in a region of operation, it might be necessary to establish more than one team assigned to the tasks. The teams act to accomplish the tasks to which they are assigned. A team is a collection of robots with similar background information. In this context, the background information is defined in the following sub-section.

3.1. Mission vector of the robots in a mobile SN

The background information of the mobile sensors (MSs) in a mobile SN is the mission vectors (MVs) embedded into them. This can be any type of information representing the characteristics of a robot. For example, it may be various attributes assigned to it such as a mission or a task, a friendship code, gender code, or any kind of nature given to it. The background information represented as a vector is defined as

$$\boldsymbol{x}_i = [x_{i1} \quad x_{i2} \quad \cdots x_{im-1} \quad x_{im}]^T, \quad i = 1, \ldots, n \tag{6}$$

In this work, we define the components of a MV in two ways. One is the simple case where the components of a MV are the weights of the background information. In this representation, the skill ratings and priorities of tasks play an important role for constituting the MV. We call this method as "MV with Skill rating-Priority". The representation of the MV takes into account the average skill ratings of all of the robots within the region of operation. We call this methods as "MV with the most skillful robot".

3.1.1. MV with skill rating – priority (SP) method

This method is based on representing the MVs by the skill ratings of the robots and the priorities of the tasks. A mission vector is the background information of the robot. The background information (weight) is the assigned missions to the mobile sensor units. As you remember, the mission has two elements, namely the skill ratings of the robots and the priorities of the tasks or targets. That is, the components of the MV are the weights,

$$x_{ij} = w_{ij} = s_i + p_j, \tag{7}$$

of the ith robot, for $j = 1, \ldots, m$. Where m is the number of tasks in the region of operation. The boundaries for the components of the vector $\boldsymbol{x}_i$ can be found as follows. Any component of $\boldsymbol{x}_i$ can be written from Eqs. (3) and (4) as

$$x_{ij} = \frac{1}{d_{ij}} + \frac{v_j}{c_j \cdot n} \tag{8}$$

Eq. (8) can be written as

$$x_{ij} = \frac{d_{ij} v_j + c_j}{d_{ij} c_j} \tag{9}$$

Now, as $d_{ij} \to \infty$,

$$x_{ij} = \frac{v_j}{c_j \cdot n} \tag{10}$$

That means that as the distance increases, the dominant parameter determining the task is the number of robots in the target. As $d_{ij} \to 0$,

$$x_{ij} = \frac{1}{d_{ij}} \to \infty \tag{11}$$

This means that as the distance decreases, the dominant term is the skill ratings of the robot.

3.1.2. MV with the Most Skillful Robot (MSR) method

In this method, the MV is the difference between the skill rating of the robot and the task minus the mean of the skill ratings between that robot and all the tasks [12]:

$$x_{ij} = s_{ij} - \sum_{k=1}^{m} \frac{s_{ik}}{m} \tag{12}$$

where m is the number of tasks. The rationale is that the robots are more likely to win tasks that have a low skill ratings for the rest of the team, but a relatively high skill rating for itself.

In this method, assume that the robot r_k has won the tasks t_i and t_j. The robot r_k will keep t_i if and only if

$$\left(s_{ki} - \sum_{l=1}^{n} \frac{s_{li}}{n}\right) > \left(s_{kj} - \sum_{l=1}^{n} \frac{s_{lj}}{n}\right) \tag{13}$$

The meaning of Eq. (13) is that the skill rating of the robot winning the task is greater than the mean of the skill ratings of all the others. That is, the robot chooses the task taking into account the team members. The task chosen actually is the best for the team not just for itself.

In this work, we choose a two dimensional vector for the simplicity to apply to the proposed concept. That is,

$$\boldsymbol{x}_i = [x_{11} \quad x_{12}]^T \tag{14}$$

The first component represents the primary mission and the second component represents the secondary mission. They are automatically updated during the course of action based on the algorithm given above. If $x_1 > x_2$, then the mobile sensor performs the first mission, otherwise it performs the second mission. Classification of the mobile sensors as separate teams is based on their missions. If the mobile sensor has to do the first (second) mission, then it must belong to the first (second) class (i.e., team). Therefore, the background information can give an opportunity to classify the mobile sensors as mission teams. Any MS

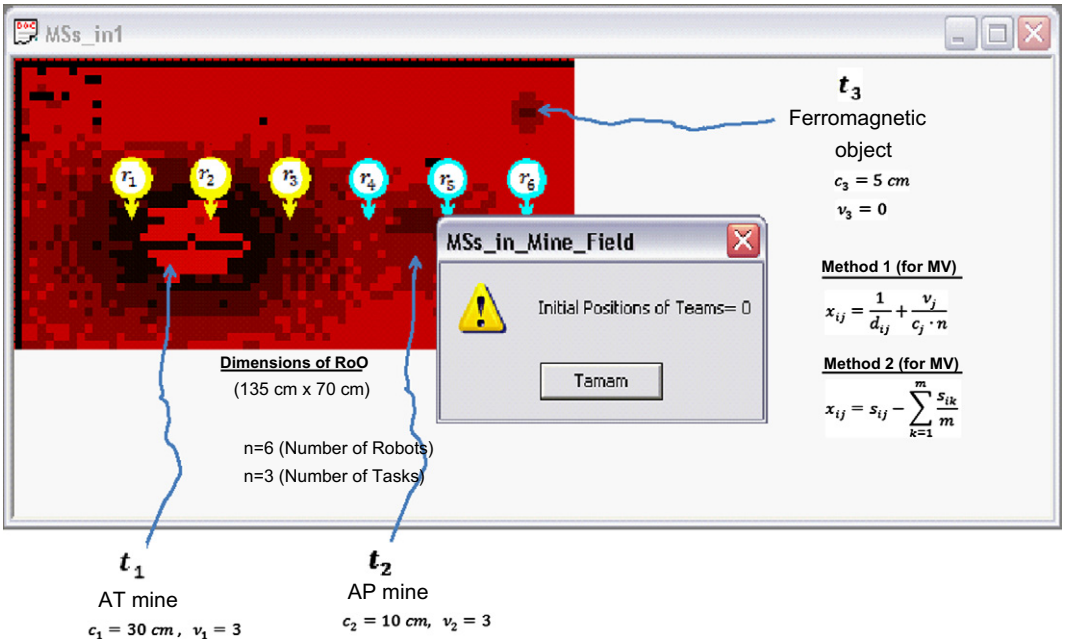


Fig. 5. Demonstration of initial assignment of mission vectors (MVs) based on two methods. The total number of robots $n = 6$, total number of tasks $m = 3$.

can do either a primary mission assigned to it or a secondary mission assigned to it. That is, classification of the mobile sensors as mission teams is based on these vectors.

As an example, we can demonstrate the methods by using one of our simulation results depicted in Fig. 5. In this simulation, six robots in total are available in the region of operation where one anti-tank (AT) mine, one anti-personnel (AP) mine and one ferromagnetic object are buried. The image is the region of operation obtained by scanning it by the magneto-resistive sensors mounted on the scanner simulating the behavior of the robots. Initially, the MVs are updated on the robots databases embedded inside them. The Table 1 gives the data needed for the analysis.

The data in Table 1 is very important. In the application, we discard the third task. So, the MV has two dimensions. Let's repeat the MVs here again and see the team constitutions clearly. For the first method where the MV creation is based on the Eq. (8), we have the following MVs (see also the Table 1):

$$\boldsymbol{x}_1 = [x_{11} \quad x_{12}]^T = [0.066 \quad 0.064]^T$$
$$\boldsymbol{x}_2 = [x_{21} \quad x_{22}]^T = [0.0826 \quad 0.07]^T$$
$$\boldsymbol{x}_3 = [x_{31} \quad x_{32}]^T = [0.0526 \quad 0.08]^T$$
$$\boldsymbol{x}_4 = [x_{41} \quad x_{42}]^T = [0.0386 \quad 0.116]^T$$
$$\boldsymbol{x}_5 = [x_{51} \quad x_{52}]^T = [0.0316 \quad 0.15]^T$$
$$\boldsymbol{x}_6 = [x_{61} \quad x_{62}]^T = [0.0286 \quad 0.086]^T$$

Note that the robots r_1 and r_2 go to t_1, and the robots r_3, r_4, r_5 and r_6 go to t_2. This can be understood from the components of MVs. When the condition, xi1 > xi2, is satisfied, the robot ri will go to t1, otherwise it will go t2.

Similarly, we can look at the results of the Method 2 where the MVs are created based on the Eq. (12). We have the following MVs (see also the Table 1):

$$\boldsymbol{x}_1 = [x_{11} \quad x_{12}]^T = [0.0253 \quad -0.0106]^T$$
$$\boldsymbol{x}_2 = [x_{21} \quad x_{22}]^T = [0.033 \quad -0.013]^T$$
$$\boldsymbol{x}_3 = [x_{31} \quad x_{32}]^T = [0.0086 \quad 0.0027]^T$$
$$\boldsymbol{x}_4 = [x_{41} \quad x_{42}]^T = [-0.014 \quad 0.03]^T$$
$$\boldsymbol{x}_5 = [x_{51} \quad x_{52}]^T = [-0.036 \quad 0.0484]^T$$
$$\boldsymbol{x}_6 = [x_{61} \quad x_{62}]^T = [-0.023 \quad 0.0007]^T$$

Note that in this case the robots r_1, r_2 and r_3 go to t_1, and the robots r_4, r_5 and r_6 go to t_2. This can also be understood from the components of MVs. When the condition, $x_{i1} > x_{i2}$, is satisfied, the robot r_i will go to t_1, otherwise it will go t_2. Notice that r_3 is at the boundary of t_1 and t_2. The components of its MV converge to the values of robots who approach to t_2. Although they are so small compared to the values of the robots of t_1, it is still in t_1. However, it may go either to t_1 or t_2. We can easily interpret that both of the methods for the constitution of MVs functions very well.

The distribution of mission vectors inside a 2-dimensional space is shown in Fig. 6. As seen in Fig. 6, the MVs are the vectors distributed inside circular regions. Fig. 6 shows the probability distribution function of the MVs

$$\boldsymbol{P}(x) = P(x_{11}, x_{12}) = \boldsymbol{A}e^{-\left[\frac{(x_{11}-x_{110})^2}{2\sigma_{x11}^2} + \frac{(x_{12}-x_{120})^2}{2\sigma_{x12}^2}\right]} \tag{15}$$

Fig. 7 illustrates the two dimensional space of the MVs and their distributions. In this example, there are two classes of teams created for this application. The densities of

Table 1
Initial assignment of mission vectors (MVs).

Distance (cm)						Skill ratings						Mission vectors (MVs) Method 1 $x_{ij} = \frac{1}{d_{ij}} + \frac{v_j}{c_j \cdot n}$				Mission vectors (MVs) Method 2 $x_{ij} = s_{ij} - \sum_{k=1}^{m} \frac{s_{ik}}{m}$			
d_{11}	20	d_{12}	70	d_{13}	96	s_{11}	0.05	s_{12}	0.014	s_{13}	0.01	x_{11}	0.066	x_{12}	0.064	x_{11}	0.025	x_{12}	−0.01
d_{21}	15	d_{22}	50	d_{23}	78	s_{21}	0.066	s_{22}	0.02	s_{23}	0.013	x_{21}	0.082	x_{22}	0.07	x_{21}	0.033	x_{22}	−0.013
d_{31}	28	d_{32}	33	d_{33}	60	s_{31}	0.036	s_{32}	0.03	s_{33}	0.016	x_{31}	0.052	x_{32}	0.08	x_{31}	0.008	x_{32}	0.0027
d_{41}	45	d_{42}	15	d_{43}	43	s_{41}	0.022	s_{42}	0.066	s_{43}	0.02	x_{41}	0.038	x_{42}	0.116	x_{41}	−0.01	x_{42}	0.03
d_{51}	65	d_{52}	10	d_{53}	25	s_{51}	0.015	s_{52}	0.1	s_{53}	0.04	x_{51}	0.031	x_{52}	0.15	x_{51}	−0.03	x_{52}	0.0484
d_{61}	83	d_{62}	28	d_{63}	17	s_{61}	0.012	s_{62}	0.036	s_{63}	0.058	x_{61}	0.028	x_{62}	0.086	x_{61}	−0.02	x_{62}	0.0007

the vectors as concentric circles in the 2-dimensional space are illustrated in Table 2 and Fig. 7. As seen, the vectors are concentrated in the circle with $r \leqslant 0.3$ units at some instant of time (Figs. 6 and 7).

3.2. Support Vector Machine (SVM)

In this paper, the robots of mobile SN are aimed to be separated into different teams. This separation is based on the missions assigned to the robots. The missions of robots can constitute a part of their genetic infrastructure. Their genetic infrastructures can be implemented as the mission vectors as described in Section 3. The mobile sensors (MSs) with similar background information (that is, mission vectors which are close to each other) come together to form a mission team. Therefore, any MSs must identify the other MSs based on their background information. For the classification of background information, a simple method called support vector machine (SVM) is used [17]. SVMs are a set of related supervised learning methods used for classification. In simple words, given a set of training examples, each marked as belonging to one of the two categories, an SVM training algorithm builds a model that predicts whether a new example falls into one category or the other [17–21]. This method well suits to our application. A hyperplane, H, can easily be determined for separating the regions where the background genetic information vectors are concentrated. It is noted that this hyperplane is an optimum separation hyperplane. In Fig. 8, there are two classes of vectors, namely, Class 1 and Class2. The vectors, $\boldsymbol{sv}_1$ and $\boldsymbol{sv}_2$ are the support vectors. H_1 and H_2 are the hyperplanes passing through the support vectors $\boldsymbol{sv}_1$ and $\boldsymbol{sv}_2$ respectively and parallel to the separation hyperplane H. H_1 identifies the class 1 and H_2 identifies class 2.

The vector $\boldsymbol{w}$ determines the hyperplane H. Notice that $\boldsymbol{w}$ is perpendicular to H. The components of the vector $\boldsymbol{w}$ seen in Fig. 8 will be determined by the Example D given below.

Consider the classification of two classes of vectors that are linearly separable (see Fig. 7). The linear classifier is the hyperplane H:

$$\boldsymbol{w}^T \cdot \boldsymbol{x} + b = 0 \tag{16}$$

with the maximum width (distance between the hyperplanes H_1 and H_2 drawn by dotted lines). The first term is the dot product of two vectors $\boldsymbol{w}$ and $\boldsymbol{x}$. The second term is a scalar variable, b, representing the shift of the hyperplane from the origin of the reference frame. We have to find the set of pairs $(\boldsymbol{w}, b)$ that characterizes the linear classifier satisfying:

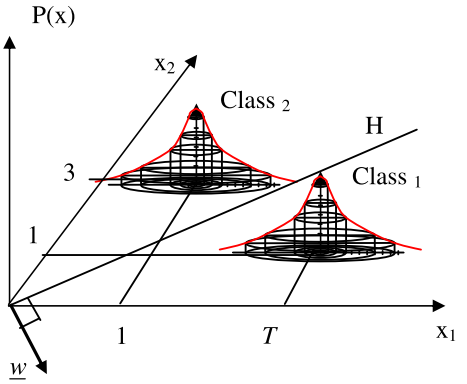


Fig. 7. Distribution of the mission vectors in hyperspace. *Legend*: *H*: Hyperplane separating two classes; *w*: The normal vector of the hyperplane.

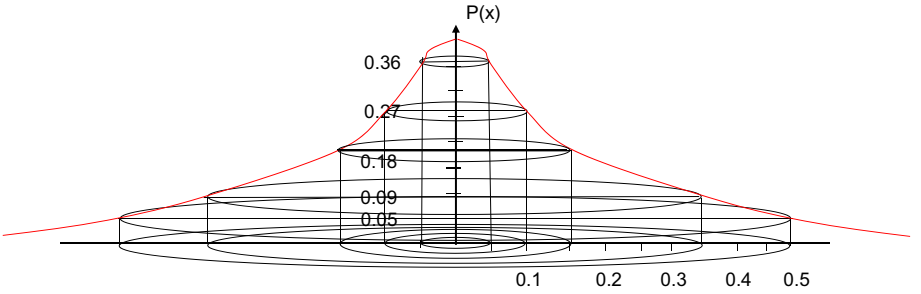


Fig. 6. Probability distribution of classification vectors in 2-dimensional space.

Table 2
Mission vector distribution.

Class 1 (Team 1)			Class 2 (Team 2)		
Diameter	No. of vectors	Probability	Diameter	No. of vectors	Probability
⩽0.10	20	0.36	⩽0.10	21	0.38
⩽0.22	15	0.27	⩽0.22	16	0.29
⩽0.30	10	0.18	⩽0.30	9	0.16
⩽0.70	5	0.09	⩽0.70	4	0.07
⩽0.95	3	0.0054	⩽0.95	2	0.037
>1.0	2	0.0036	>1.0	2	0.037

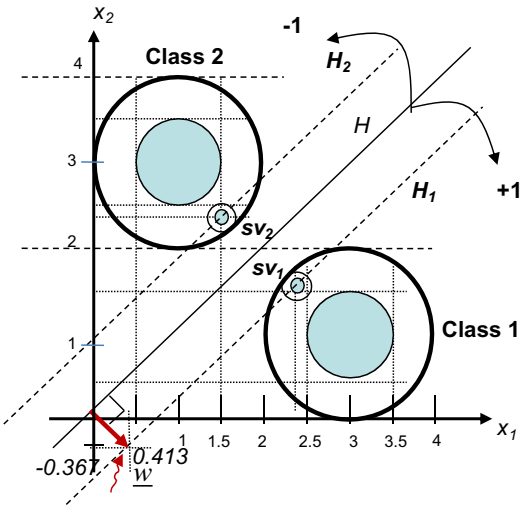


Fig. 8. The distribution of vectors inside a 2-dimensional space. *Legend*: sv_i: Support vectors ($i = 1, 2$); H: Hyperplane; w: Normal vector to the hyperplane representing it; x_i: ith coordinate of the space; H_i: Hyperplane passing through the support vector sv_i.

$$y_i = \boldsymbol{w}^T \cdot \boldsymbol{x} + b \tag{17}$$

$$y_o = c_i y_i \geqslant 1 \tag{18}$$

where

$$c_i \in \{-1, \quad 1)\} \tag{19}$$

A Lagrangian is given as

$$f = \frac{1}{2}\|\boldsymbol{w}\|^2 + \sum_{i=1}^{n} \alpha_i (c_i y_i - 1) \tag{20}$$

where α_i's are non-negative Lagrange multipliers. We must now minimize f given in Eq. (21) with respect to $\boldsymbol{w}$ and b simultaneously. This requires that the derivatives of f with respect to all the α_i's vanish, all subject to the constraints $\alpha_i \geqslant 0$. Now, this is a convex quadratic programming problem, since the objective function is itself convex, and those parts that satisfy the constraints also form a convex set. The solution can be expressed in terms of linear combination of the training vectors as

$$\boldsymbol{w} = \sum_{i=1}^{n} \alpha_i c_i \boldsymbol{x}_{svi} \tag{21}$$

It is known that the square of the norm of a vector is

$$\|\boldsymbol{w}\|^2 = \boldsymbol{w}^T \cdot \boldsymbol{w} \tag{22}$$

By putting Eq. (22) into Eq. (21),

$$f = \frac{3}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j c_i c_j \boldsymbol{x}_{svi}^T \cdot \boldsymbol{x}_{svj} + \left(\sum_{i=1}^{n} (\alpha_i c_i) \cdot b - \sum_{i=1}^{n} \alpha_i \right. \tag{23}$$

Minimize f with respect to α_i and b subject to

$$\alpha_i \geqslant 0 \tag{24}$$

And

$$\sum_{i=1}^{n} \alpha_i c_i = 0 \tag{25}$$

That is,

$$\frac{df}{d\alpha_i} = 0, \quad i = 1, 2, \ldots, n \tag{26}$$

$$\frac{df}{db} = 0 \tag{27}$$

For $n = 2$,

$$f = \frac{3}{2}\alpha_1 \alpha_1 c_1 c_1 \boldsymbol{x}_{sv1}^T \cdot \boldsymbol{x}_{sv1} + 3\alpha_1 \alpha_2 c_1 c_2 \boldsymbol{x}_{sv1}^T \cdot \boldsymbol{x}_{sv2} + \frac{3}{2}\alpha_2 \alpha_2 c_2 c_2 \boldsymbol{x}_{sv2}^T \cdot \boldsymbol{x}_{sv2} + \alpha_1 c_1 b + \alpha_2 c_2 b - \alpha_1 - \alpha_2 \tag{28}$$

Minimize f:

$$\frac{df}{d\alpha_1} = 3\alpha_1 c_1 c_1 \boldsymbol{x}_{sv1}^T \cdot \boldsymbol{x}_{sv1} + 3\alpha_2 c_1 c_2 \boldsymbol{x}_{sv1}^T \cdot \boldsymbol{x}_{sv2} + c_1 b - 1 = 0 \tag{29}$$

$$\frac{df}{d\alpha_2} = 3\alpha_1 c_1 c_2 \boldsymbol{x}_{sv1}^T \cdot \boldsymbol{x}_{sv2} + 3\alpha_2 c_2 c_2 \boldsymbol{x}_{sv1}^T \cdot \boldsymbol{x}_{sv2} + c_2 b - 1 = 0 \tag{30}$$

$$\frac{df}{db} = \alpha_1 c_1 + \alpha_2 c_2 = 0 \tag{31}$$

Find

$$\alpha_1 = \alpha_2 \quad \text{and } b \text{ for} \quad c_1 = -1, \quad c_2 = +1, \quad \boldsymbol{x}_{sv1} = [x_{11} \quad x_{21}]^T, \boldsymbol{x}_{sv2} = [x_{12} \quad x_{22}]^T.$$

After finding the Lagrange multipliers satisfying above rules, an optimum hyperplane H can be plotted perpendicular to the weighting vector $\boldsymbol{w} = \sum_{i=1}^{n} \alpha_i c_i \boldsymbol{x}_{svi}$ (see Eq. (22)). The classification can be achieved by the dot product of any vector $\boldsymbol{x}$ to be classified by the vector $\boldsymbol{w}$, that is, $y = \boldsymbol{w}^T \cdot \boldsymbol{x}$. If $y > +1$, x belongs to class 1 else it belongs to class 2.

3.3. Example

Let's the support vectors be

$\boldsymbol{x}_{sv1} = [1.5 \quad 2.3]^T$, $\boldsymbol{x}_{sv2} = [2.4 \quad 1.5]^T$, $c_1 = -1$ and $c_2 = 1$

Find the hyperplane that can separate the two dimensional space into two classification regions. The Eq. (24) can be rewritten as

$$f = \frac{3}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j c_i c_j \underline{x}_{svi}^T \cdot x_{svj} + \left(\sum_{i=1}^{n}\alpha_i c_i\right)\cdot b - \sum_{i=1}^{n}\alpha_i$$

For $n = 2$,

$$f = \frac{1}{2}[\alpha_1 c_1 \underline{x_1}^T + \alpha_2 c_2 \underline{x_2}^T]\cdot[\alpha_1 c_1 \underline{x_1} + \alpha_2 c_2 \underline{x_2}] + \alpha_1(c_1[\alpha_1 c_1 \underline{x_1}^T + \alpha_2 c_2 \underline{x_2}^T]\cdot \underline{x_1} + b] - 1) + \alpha_2(c_2[\alpha_1 c_1 \underline{x_1}^T + \alpha_2 c_2 \underline{x_2}^T]\cdot \underline{x_2} + b] - 1)$$

It can be reduced to Eq. (28):

$$f = \frac{3}{2}\alpha_1\alpha_1 c_1 c_1 \boldsymbol{x}_{sv1}^T \cdot \boldsymbol{x}_{sv1} + 3\alpha_1\alpha_2 c_1 c_2 \boldsymbol{x}_{sv1}^T \cdot \boldsymbol{x}_{sv2} + \frac{3}{2}\alpha_2\alpha_2 c_2 c_2 \boldsymbol{x}_{sv2}^T \cdot \boldsymbol{x}_{sv2} + \alpha_1 c_1 b + \alpha_2 c_2 b - \alpha_1 - \alpha_2$$

From Eqs. (29)–(31) and putting the support vectors,

$$22.62\alpha_1 - 21.15\alpha_2 - b = 1 \tag{32}$$

$$-21.15\alpha_1 + 24.03\alpha_2 + b = 1 \tag{33}$$

$\frac{df}{db} = \alpha_1 c_1 + \alpha_2 c_2 = 0$ gives $-\alpha_1 + \alpha_2 = 0$, that is, $\alpha_1 = \alpha_2$. Putting this result into Eqs. (33) and (34) gives

$$\alpha_1 = \alpha_2 = 0.45977 \tag{34}$$

and

$$b = -0.3241 \tag{35}$$

Then the $\boldsymbol{w}$ vector can be written as (see Fig. 8)

$$\boldsymbol{w} = \sum_{i=1}^{2}\alpha_i c_i \boldsymbol{x}_{svi} = [0.413793 \quad -0.367816]^T \tag{36}$$

3.4. Implementation of SVM for team establishment

The SVM classifier can easily be implemented with operational amplifiers and logic gates as shown in Fig. 9. In this application, the mission vectors are 2-dimensional as explained above. This means that there are two classes of MSs in the operation area. It receives the class vectors as inputs. The input stage, which is a summing amplifier, implements the dot product of the input vector with the vector $\boldsymbol{w}$. It then produces the output signal as $y_i = \boldsymbol{w}^T \cdot x_i$. The next stage is a limiter that decides whether y_0 is -1 or $+1$ representing the appropriate class to which it

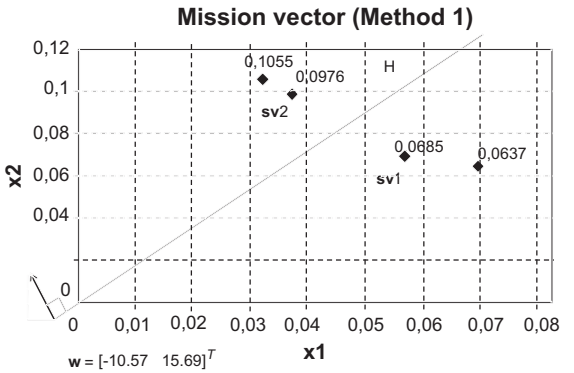


Fig. 10. The hyperplane H separating two teams of robots. Here, the mission vectors are $\boldsymbol{x}_1 = [0.06926 \quad 0.0637]^T$, $\boldsymbol{x}_2 = [0.0566 \quad 0.0685]^T$, $\boldsymbol{x}_3 = [0.037 \quad 0.0976]^T$, and $\boldsymbol{x}_4 = [0.0317 \quad 0.1055]^T$. The support vectors are chosen as $\boldsymbol{x}_2 = sv_1 = [0.0566 \quad 0.0685]^T$ and $\boldsymbol{x}_3 = \boldsymbol{sv}_2 = [0.037 \quad 0.0976]^T$.

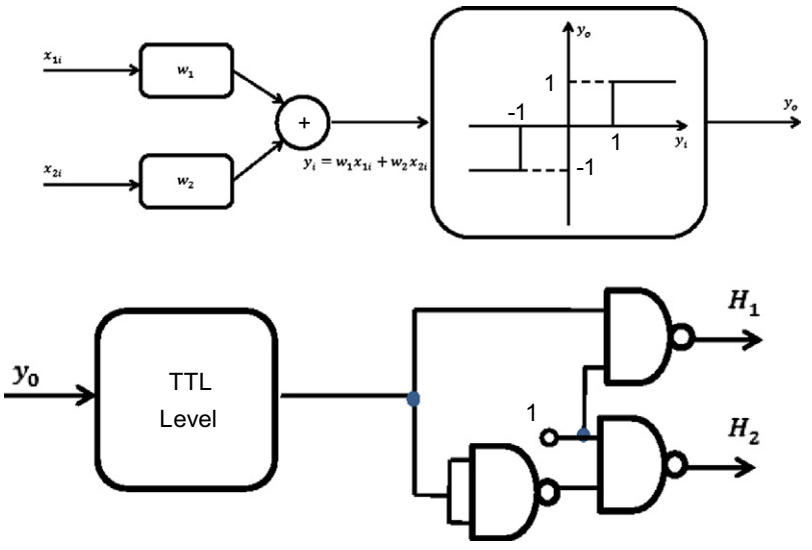


Fig. 9. SVM classifier for team establishment.

Table 3
Initial assignment of mission vectors (MVs).

Distance (cm)						Skill ratings						Mission vectors (MVs) Method 1 $x_{ij} = \frac{1}{d_{ij}} + \frac{y_j}{c_j \cdot n}$				Mission vectors (MVs) Method 2 $x_{ij} = s_{ij} - \sum_{k=1}^{m} \frac{s_{ik}}{m}$			
d_{11}	19	d_{12}	73	d_{13}	100	s_{11}	0.052	s_{12}	0.014	s13	0.01	x_{11}	0.069	x_{12}	0.064	x_{11}	0.019	x_{12}	−0.019
d_{21}	25	d_{22}	54	d_{23}	78	s_{21}	0.040	s_{22}	0.018	s23	0.013	x_{21}	0.056	x_{22}	0.07	x_{21}	0.010	x_{22}	−0.010
d_{31}	49	d_{32}	21	d_{33}	40	s_{31}	0.020	s_{32}	0.048	s33	0.025	x_{31}	0.037	x_{32}	0.098	x_{31}	−0.013	x_{32}	0.013
d_{41}	66	d_{42}	18	d_{43}	25	s_{41}	0.015	s_{42}	0.055	s43	0.040	x_{41}	0.031	x_{42}	0.105	x_{41}	−0.02	x_{42}	0.02

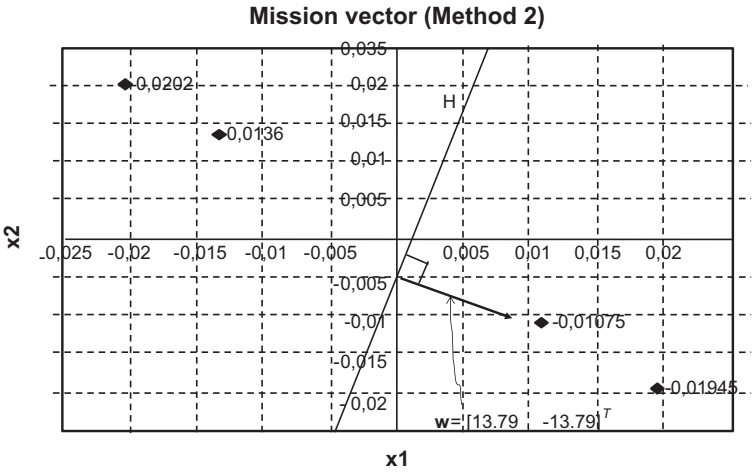


Fig. 11. The hyperplane H separating two teams of robots. Here, the mission vectors are $\boldsymbol{x}_1 = [0.01945 \quad -0.01945]^T$, $\boldsymbol{x}_2 = [0.01075 \quad -0.01075]^T$, $\boldsymbol{x}_3 = [-0.0136 \quad 0.0136]^T$ and $\boldsymbol{x}_4 = [0.0202 \quad 0.0202]^T$. The support vectors are chosen as $\boldsymbol{x}_3 = \boldsymbol{sv}_1 = [-0.0136 \quad 0.0136]^T \cdot \boldsymbol{x}_2 = \boldsymbol{sv}_2 = [0.01075 \quad -0.01075]^T$.

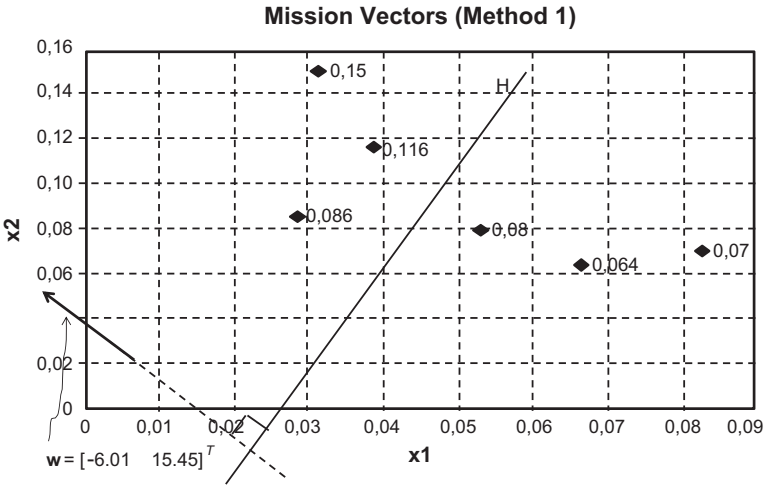


Fig. 12. The hyperplane H separating two teams of robots. Here, the mission vectors are $\boldsymbol{x}_1 = [0.0660 \quad 0.0640]^T$, $\boldsymbol{x}_2 = [0.0826 \quad 0.0700]^T$, $\boldsymbol{x}_3 = [0.0526 \quad 0.0800]^T$, $\boldsymbol{x}_4 = [0.0386 \quad 0.1160]^T$, $\boldsymbol{x}_5 = [0.0316 \quad 0.1500]^T$ and $\boldsymbol{x}_6 = [0.0286 \quad 0.0860]^T$. The support vectors are chosen as $\boldsymbol{x}_3 = \boldsymbol{sv}_1 = [0.0526 \quad 0.0800]^T$ and $\boldsymbol{x}_4 = \boldsymbol{sv}_2 = [0.0386 \quad 0.1160]^T$.

belongs. Since a TTL circuit is used to activate the corresponding gate output, it is first converted to the TTL level and then it is passed through the logic circuit realized by the NAND gates. The outputs H_1 and H_2 are active low. That is, if $H_i = 0$, then the robot is a member of ith team (where $i = 1, 2$).

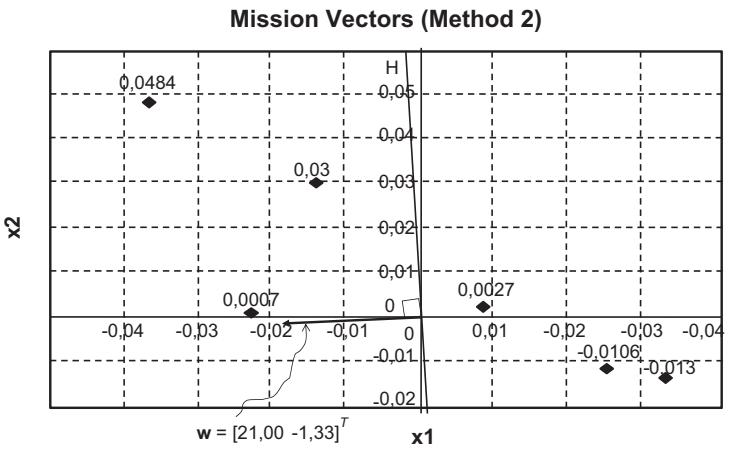


Fig. 13. The hyperplane H separating two teams of robots. Here, the mission vectors are $\boldsymbol{x}_1 = [0.0253 \quad -0.0106]^T$, $\boldsymbol{x}_2 = [0.0330 \quad -0.0130]^T$, $\boldsymbol{x}_3 = [0.0086 \quad 0.0027]^T$, $\boldsymbol{x}_4 = [-0.0140 \quad 0.0300]^T$, $\boldsymbol{x}_5 = [-0.0366 \quad 0.0484]^T$ and $\boldsymbol{x}_6 = [-0.0230 \quad 0.0007]^T$. The support vectors are chosen as $\boldsymbol{x}_3 = \boldsymbol{sv}_1 = [0.0086 \quad 0.0027]^T$ and $\boldsymbol{x}_6 = \boldsymbol{sv}_2 = [-0.023 \quad 0.0007]^T$.

Fig. 14. Simulation with four mobile sensors grouped automatically into two teams in a magnetic field environment.

4. Experimental results

The operation of the system is tested by simulations implemented in Visual C++ environment and also an experimental prototype systems realized as experimental Sumo robots. The magnetic anomalies are created by one real anti-tank (AT) mine, one training anti-personnel (AP) mine and a ferromagnetic object (a bolt). For illustrative purposes, some of the simulations and practical application results are given in Figs. 14–17. The locations of anomalies created by the AT mine, AP mine and the ferromagnetic bolt are considered as the first task, the second task and the third task respectively. The task assignments and team establishments processes for each case are sum-

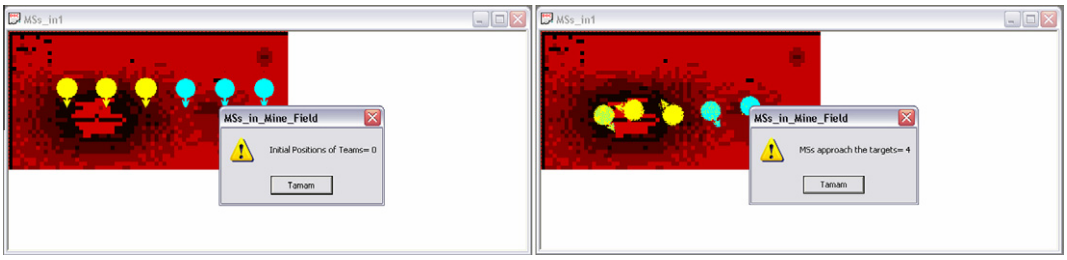


Fig. 15. Simulation with six mobile sensors grouped automatically into two teams in a magnetic field environment.

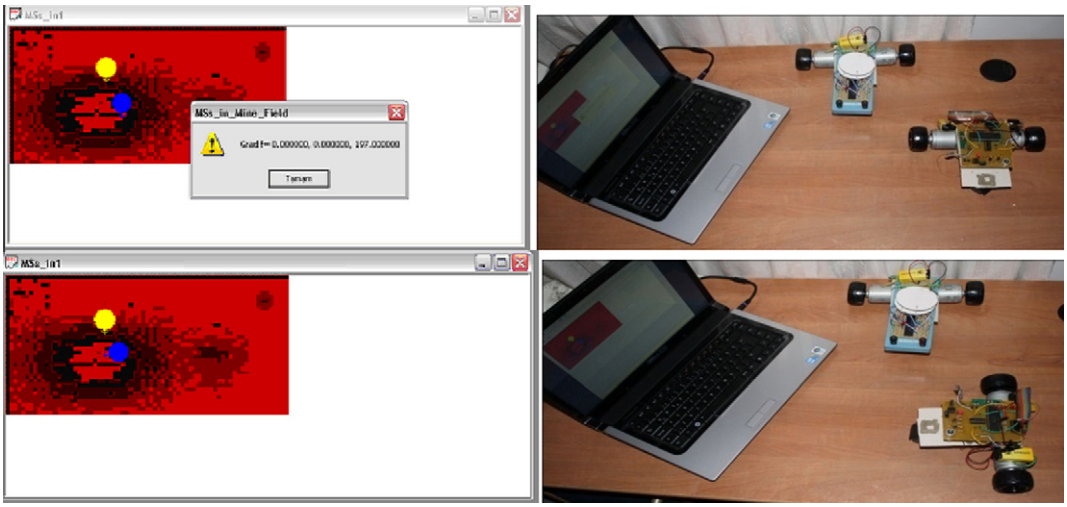


Fig. 16. Experiments with prototype experimental robots. In these pictures, the two robots detect the anomalies created under the wooden platforms and the motions are monitored on the computer screen.

marized in Tables 1–3. In the experiments, the anomalies of the AT and AP mines are taken as the tasks and the ferromagnetic object is discarded in order to obtain a two-dimensional mission vectors.

The experiments are carried out by using the following parameters: the diameters of the first job assignment circles are $c_1 = 30$ cm, $c_2 = 10$ cm which are the diameters of AT and AP mines respectively. The two methods, method 1 (MV with Skill rating – Priority (SP) Method) and method 2 (MV with the Most Skillful Robot (MSR) Method) are used for the determination of the mission vectors.

For the first experiment (Fig. 10), the number of robots $n = 4$, the number of tasks $m = 2$, the number of robots at the vicinity of task 1 is $\vartheta_1 = 2$, and the number of robots at the vicinity of task 2 is $\vartheta_2 = 2$. Table 3 summarizes the results of the first experiment.

From Table 3, the mission vectors obtained using method 1 (MV with Skill rating – Priority (SP) Method) are as follows: $\boldsymbol{x}_1 = [0.06926 \quad 0.0637]^T$, $\boldsymbol{x}_2 = [0.0566 \quad 0.0685]^T$, $\boldsymbol{x}_3 = [0.037 \quad 0.0976]^T$, and $\boldsymbol{x}_4 = [-0.00317 \quad 0.1055]]^T$. The support vectors are chosen as $\boldsymbol{x}_2 = \boldsymbol{sv}_1 = [0.0566 \quad 0.0685]^T$ and $\boldsymbol{x}_3 = \boldsymbol{sv}_2 = [0.037 \quad 0.0976]^T$.

Using Eqs. (29)–(31), the non-negative Lagrange multipliers can be calculated as $\alpha_1 = \alpha_2 = 539.23$ and the offset value $b = 2.435$. Using Eq. (21), the vector $\boldsymbol{w}$ can be calculated as $\boldsymbol{w} = [-10.57 \quad 0.1055]^T$. This vector defines the hyperplane H separating two classes of mission vectors that is used for the establishment of the teams (Fig. 10).

The mission vectors calculated by the method 2 (MV with the Most Skillful Robot (MSR) Method) are as follows: $\boldsymbol{x}_1 = [0.01945 \quad -0.01945]^T$, $\boldsymbol{x}_2 = [0.01075 \quad -0.01075]^T$, $\boldsymbol{x}_3 = [-0.0136 \quad 0.0136]^T$, and $\boldsymbol{x}_4 = [-0.0202 \quad 0.0202]^T$. For these mission vectors, the hyperplane is shown in Fig. 11 for which $\boldsymbol{w} = [13.79 \quad -13.79]^T$ and $b = 0.116$.

For the second illustrative experiment (Fig. 14), $n = 6$, $m = 2$, $\vartheta_1 = 3$, $\vartheta_2 = 3$. The results of the task assignment process are given in Table 1.

From Table 1, the mission vectors obtained using method 1 (MV with Skill rating – Priority (SP) Method) are as follows: $\boldsymbol{x}_1 = [0.0660 \quad 0.0640]^T$, $\boldsymbol{x}_2 = [0.0826 \quad 0.0700]^T$, $\boldsymbol{x}_3 = [0.0526 \quad 0.0800]^T$, $\boldsymbol{x}_4 = [0.0386 \quad 0.1160]^T$, $\boldsymbol{x}_5 = [0.0316 \quad 0.1500]^T$ and $\boldsymbol{x}_6 = [0.0286 \quad 0.860]^T$. The support vectors are chosen as $\boldsymbol{x}_3 = \boldsymbol{sv}_1 = [0.0526 \quad 0.0800]^T$ and $\boldsymbol{x}_4 = \boldsymbol{sv}_2 = [0.0386 \quad 0.1160]^T$.

Using Eqs. (29)–(31), the non-negative Lagrange multipliers can be calculated as $\alpha_1 = \alpha_2 = 429.38$ and the offset value $b = -3.72$. Using Eq. (21), the vector $\boldsymbol{w}$ can be calculated as $\boldsymbol{w} = [-6.01 \quad 15.45]^T$. This vector defines the hyperplane H separating two classes of mission vectors that is used for the establishment of the teams (Fig. 12).

The mission vectors calculated by the method 2 (MV with the Most Skillful Robot (MSR) Method) are as follows:

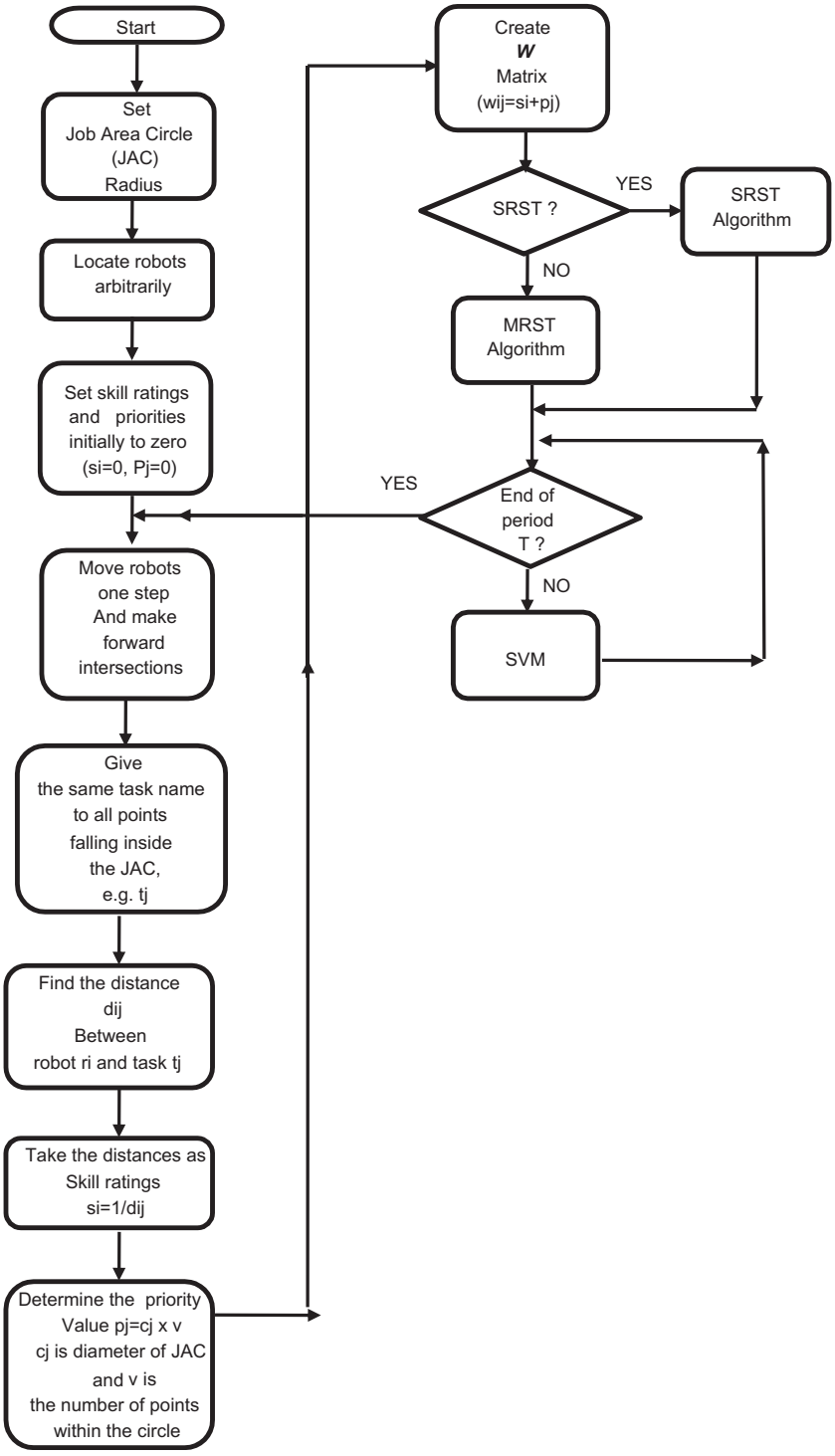


Fig. 17. The flow chart of the overall operation.

$\boldsymbol{x}_1 = [0.0253 \quad -0.0106]^T$, $\boldsymbol{x}_2 = [0.0330 \quad -0.0130]^T$, $\boldsymbol{x}_3 = [0.0086 \quad 0.0027]^T$, $\boldsymbol{x}_4 = [-0.0140 \quad 0.0300]^T$, $\boldsymbol{x}_5 = [-0.0366 \quad 0.0484]^T$ and $\boldsymbol{x}_6 = [-0.0230 \quad 0.0007]^T$. The support vectors are chosen as $\boldsymbol{x}_3 = \boldsymbol{sv}_1 = [0.0086 \quad 0.0027]^T$ and $\boldsymbol{x}_6 = \boldsymbol{sv}_2 = [-0.023 \quad 0.0007]^T$.

Using Eqs. (29)–(31), the non-negative Lagrange multipliers can be calculated as $\alpha_1 = \alpha_2 = 664.96$ and the offset value $b = -0.447$. Using Eq. (21), the vector $\boldsymbol{w}$ can be calculated as $\boldsymbol{w} = [-21.00 \quad -1.33]^T$. This vector defines the hyperplane H separating two classes of mission

vectors that is used for the establishment of the teams (Fig. 13).

And for the last experiment performed by prototype robots, $n = 2$, $m = 2$, $\vartheta_1 = 1$, $\vartheta_2 = 1$. In this experiment, we tried to see the functioning and performance of the practical system in performing task assignment and team establishment by SVM, though the number of robots are very restricted.

And for the last experiment performed by prototype robots, $n = 2$, $m = 2$, $\vartheta_1 = 1$, $\vartheta_2 = 1$. In this experiment, we tried to see the functioning and performance of the practical system in performing task assignment and team establishment by SVM, though the number of robots are very restricted.

5. Conclusion

In this paper, the team establishment problem is solved by using task assignment approach. A new algorithm is developed for task assignment. The task assignment algorithm has two versions. The first version is used for the assignment of single robot to a single task. However, the second version is used for the assignment of multiple robots to a single task. The algorithm depends on the optimization of weights composed of skill ratings of the robots and priorities of the tasks. The skill ratings are related to the distances from the robots to the tasks. The shorter the distance is, the more skillful the robot. The priorities of the tasks are the values mainly determined from the number of robots aiming at the target objects. It is the measure of importance of the task. A priority can be defined as the number of robots aiming to that target multiplied by the inverse of the size of the object. Two methods are used to define the weights. The weights are used for the creation of mission vectors which constitute the background information of the robots. Depending on the mission vectors, robots are classified as teams.

In this work, the operation of robot teams is performed in a periodic fashion. The period is divided into two stages. The first stage is the initialization stage where the initial task assignment is done by the task assignment algorithm developed here. In the second stage, the team establishment is achieved by using SVM method which creates the teams based on the mission vectors of the robots obtained in the first stage of the period. Dividing the course of action in this way facilitates the operation of the teams and also gives an opportunity to update the mission vectors during the operation. It reduces the communications needs as well. During the second stage, robots move without any communications. They come together as teams by use of the SVM mechanism.

The sensor network created here is a mobile sensor network that is composed of mobile units making rotational and translational motion within a period of operation. Each mobile unit has a capability of sensing the direction of the gradient vector of the magnetic field by means of a convolution mask created by a sensor mechanism with 3×3 sensor grid. The overall operation is synchronized in time by use of a time division multiple access (TDMA) method.

The methods are demonstrated by simulations and a practical example. In order to simplify the proof of the concept, a two dimensional mission vectors are used. The task assignment algorithm developed here has no restriction on the dimension of vectors. It may be applied for assignment of many robotic teams to many tasks. The SVM method is also suitable for separation of multiple regions whether linear of nonlinear fashion. In this application, the SVM approach is adapted for the classification of two dimensional vectors. In that case, a linear separation hyperplane can easily be found. But it can be extended to multiple and nonlinear separations as well.

The approach developed here gives satisfactory results. It suits very well the solution of the problem for detecting anti-tank and anti-personnel mines buried long years before at the border regions. The method helps the works intended to clear this kind of regions.

Appendix A

A.1. Illustrations

In this appendix, the outputs of team establishment process are given. The process automatically determines the skill ratings based on the distances to the tasks and priorities of the tasks. These values are used to create the mission vectors of the robots. They move in the potential field and constitute the teams based on these vectors by means of SVMs. The mission vectors are periodically updated during the course of action. The first illustration shows the behavior of four robots in a magnetic anomaly environment. In the second illustration, the number of robots is increased to six. The third illustration is the conceptual demonstration of the proposed approach by two practical experimental robotic systems. The results are encouraging (see Figs. 14–17).

A.2. Flowchart

This appendix gives the flow chart of the overall operation. It includes the team assignment algorithms as well.

References

[1] S. Nazlibilek, O. Kalender, Y. Ege, Mine identification and classification by mobile sensor network using magnetic anomaly, IEEE Transactions on Instrumentation and Measurement 60 (3) (2011) 1028–1036.
[2] Brian P. Gerkey, Maja J. Mataric, Multi-robot task allocation: analysing the complexity and optimality of key architectures, in: Proceedings of the IEEE International Conference on Robotics and Automation, Taipei, Taiwan, September 2003, pp. 3862–3868.
[3] L.E. Parker, ALLIANCE: an architecture for fault tolerant multirobot cooperation, IEEE Transactions on Robotics and Automation 14 (2) (1998) 220–240.
[4] B.B. Werger, M.J. Mataric, Broadcast of local eligibility for multi-target observations, in: Lynne E. Parker, George Bekey, Jacob BArhen (Eds.), Distributed Autonomous Robotic Systems, vol. 4, Springer Verlag, 2000. pp. 347–356.
[5] M.J. Mataric, Behavior-based control: examples from navigation, learning and group behavior, Journal of Experimental and Theoretical Artificial Intelligence 9 (2–3) (1997) 323–336.
[6] S.C. Botelho, R. Alami, M+: a scheme for multi-robot cooperation through negotiated task allocation and achievement, in: Proceedings

of the IEEE International Conference on Robotics and Automation, Detroit, 1999.
[7] B.P. Gerkey, M.J. Matarić, Murdoch: publish/subscribe task allocation for heterogeneous agents, in: Proceedings of the Fourth International Conference on Autonomous Agents, Barcelona, Spain, 2000, pp. 203–204.
[8] G. Smith, The contract net protocol: high-level communication and control in a distributed problem solver, IEEE Transactions on Computers 29 (12) (1980).
[9] M.B. Dias, R. Zlot, N. Kalra, A. Stentz, Market-based multirobot coordination: a survey and analysis, Proceedings of the IEEE Special Issue on Multirobot Systems 94 (7) (2006).
[10] R. Zlot, A. Stentz, Complex task allocation for multiple robots, in: IEEE International Conference on Robotics and Automation, 2005.
[11] Robert Zlot, Anthony Stentz, Market-based multirobot coordination for complex tasks, The International Journal of Robotics Research 25 (1) (2006) 73–101.
[12] A. Howard, A. Viguria, Controlled reconfiguration of robotic mobile sensor networks using distributed allocation formalisms, in: NASA Science Technology Conference (NSTC 2007), Maryland, USA, 2007.
[13] T.S. Dahl, M. Mataric, G.S. Sukhatme, Multi-robot task allocation through vacancy chain scheduling, Elsevier, Robotics and Autonomous Systems 57 (2009) 674–687.
[14] Guido Fioretti, A model of vacancy chains as a mechanism for resource allocation, Journal of Mathematical Sociology 34 (1) (2010) 52–75.
[15] David Gale, The Theory of Linear Economic Models, McGraw-Hill Book Company, Inc., New York, 1960.
[16] Harold W. Kuhn, The Hungarian method for the assignment problem, Naval Research Logistics Quarterly 2 (1) (1955) 83–97.
[17] C. Cortes, V. Vapnik, Support vector networks, Machine Learning 20 (1995) 273–297.
[18] Christopher J.C. Burges, A Tutorial on Support Vector Machines for Pattern Recognition, Data Mining and Knowledge Discovery, vol. 2, Kluwer Academic Publisher, Boston, 1998. pp. 121–167.
[19] V. Vapnik, Estimation of Dependences Based on Empirical Data, Nauka, Moscow, 1979. (in Russian, English translation: Springer Verlag, New York, 1982).
[20] B. Schölkopf, A. Smola, K.-R. Müller, C. Burges, V. Vapnik, Support vector methods in learning and feature extraction, Australian Journal of Intelligent Information Processing Systems 5 (1998) 3–9 (Special issue with selected papers of ACNN'98).
[21] B. Schölkopf, K. Sung, C. Burges, F. Girosi, P. Niyogi, T. Poggio, V. Vapnik, Comparing support vector machines with gaussian kernels to radial basis function classifiers, IEEE Transactions on Signal Processing 45 (1997) 2758–2765.