

Self-adaptive decision-making mechanisms to balance the execution of multiple tasks for a multi-robots team

Nunzia Palmieri^{a,*}, Xin-She Yang^b, Floriano De Rango^a, Amilcare Francesco Santamaria^a

^aDepartment of Computer Engineering, Modeling, Electronics and System Science, University of Calabria, Italy

^bDesign Engineering and Mathematics, School of Science and Technology, Middlesex University London, UK

Received 27 June 2017, Revised 22 March 2018, Accepted 27 March

Published online 3 May 2018

Abstract

This work addresses the coordination problem of multiple robots with the goal of finding specific hazardous targets in an unknown area and dealing with them cooperatively. The desired behaviour for the robotic system entails multiple requirements, which may also be conflicting. The paper presents the problem as a constrained bi-objective optimization problem in which mobile robots must perform two specific tasks of exploration and at same time cooperation and coordination for disarming the hazardous targets. These objectives are opposed goals, in which one may be favored, but only at the expense of the other. Therefore, a good trade-off must be found. For this purpose, a nature-inspired approach and an analytical mathematical model to solve this problem considering a single equivalent weighted objective function are presented. The results of proposed coordination model, simulated in a two dimensional terrain, are showed in order to assess the behaviour of the proposed solution to tackle this problem. We have analyzed the performance of the approach and the influence of the weights of the objective function under different conditions: static and dynamic. In this latter situation, the robots may fail under the stringent limited budget of energy or for hazardous events. The paper concludes with a critical discussion of the experimental results.

Keywords: Multi-Robot Systems, Bi-criteria Optimization Model, Nature-Inspired Algorithms, Self-Coordination of Multiple Robots, Hazardous Environment.

1. Introduction

Search for unknown targets and then management of the found targets can have great importance to many applications related to disaster situations like earthquakes, counter terrorist attacks, clean-up of minefield and handling of lost hazardous objects. Over the last decade, many researchers have been studying groups of robots that act as a team to accomplish certain difficult tasks in dynamic, unknown and hazardous environments. A team of robots can be assigned to a task that is too heavy and/or harmful to humans and such robots can accomplish the assigned task in a faster, cheaper, more efficient way, providing a better robustness and adaptability [1, 2].

This paper studies the coordination of multiple robots that need to explore an unknown area, in

order to detect and disarm cooperatively dangerous sources (e.g., land mines, hazardous chemicals), since it is either impossible or too expensive for only one robot to carry out the task individually. In these applications, robots must have the ability to distribute themselves among various locations of the area and also redistribute them at the target's locations, in order to handle them in parallel. Furthermore, the team is designed to complete the overall task in a cost-saving way, with an aim of the minimization of the mission time. This issue is similar to a task allocation problem, but the main difference is that the locations of targets are not known *a priori* and the assignment can be dynamic as the number of targets found can be one or more at any time.

The most common approach is instantaneous task allocation, which means that the robots are instantaneously assigned to the targets that could give the maximum benefit at the moment and hence the task allocation is achieved according a greedy

*Corresponding author

Email address: n.palmieri@dimes.unical.it (Nunzia Palmieri)

strategy. However the instantaneous task allocation does not take into account any future events and it may degrade the performance. A desirable feature of task allocation scheme, especially in a dynamic uncertain environment, is that the robots can be well-prepared to react to new events that can occur.

Therefore, the main aim of this work is to propose an approach which attempts to simultaneously minimize the exploration time and the coordination/recruitment time. The goal is to assign each robot to the best task from its point of view, balancing exploration and recruitment and at the same time reacting in an efficient manner to the events that can occur anytime. In our proposed approach, the formulation of the problem is considered as a bi-objective model with constraints and the use of nature-inspired approaches. Then, a weighted objective function is proposed to balance the two goals, and specific values of the weights are investigated in order to analyze different scenarios in the proposed solution. It is worth pointing out that we did not focus on the detailed mechanism of detection and disarmament of the targets such as the exact procedure of handling a hazardous target. Rather, we suppose that the target can be detected by proper sensors and dealing with through certain actions that can be modelled as a fixed time delay. Consequently, we are more interested in how to provide the coordinated motions in a distributed and decentralized decision mechanism when the information about the environment for each robot is only partially available or altogether absent.

However, two major problems exist: how to explore the area and make the decision on where to move effectively at a reasonable computational cost, and how to avoid deadlock; that is, the situation where robots are waiting for a long time for the others to proceed to disarming process. These issues are particularly relevant in strictly collaborative tasks since the robots need to work collectively and adaptively for the disarmament of the hazardous targets, and each robot has only locally and partially information about the environment.

The paper presents a new swarm intelligence (SI)-based approach that is strongly inspired by the biological behaviour of social insects. Biology-inspired metaheuristic algorithms have recently become the forefront of the current research as an efficient way to deal with many NP-hard combinatorial optimization problems and non-linear optimization constrained problems in general [3]. These

algorithms are based on some particular successful mechanisms of biological phenomena of mother nature in order to achieve the survival of the fittest in a dynamically changing environment. Examples of collective behaviours in nature are numerous. They are based, mainly, on direct or indirect exchange of information about the environment between the members of the swarm. Although the rules governing the interactions at the local level are usually easy to describe, the result of such behaviour is difficult to predict. However, through collaboration, swarms in nature are able to solve complex problems that are crucial for their survival.

More specifically in this paper, our work combines two bio-inspired algorithms using only local interactions between the members of the swarm and with the environment. Therefore, in our approach, each robot is able to self-organize and uses only simple information. Using this information, it assesses the task opportunities individually, and makes movements through a incremental phase. The action mechanism relies on two types of communication: indirect communication and direct communication.

First, we introduce an algorithm for the exploration of the area based on a repulsive pheromone mechanism where no Wi-Fi communication is possible. An indirect communication through depositing and sensing chemical pheromone on the grid cells is available as an exploration mechanism for the deployment of the robots in the environment. This form of communication is embedded in the environment and it is inspired by ants that lay pheromone as marking to be sensed later by others (including possibly themselves). The environment allows the aggregation of the local interactions of numerous robots in order to achieve a certain aim. The environment becomes a shared memory on which information can be stored and removed (according to certain natural mechanisms). The algorithm is inspired by the cooperative behaviour in nature, as the colonies of social insects [4].

On the other hand, when a robot detects a target, in order to handle it cooperatively and more quickly, it manages and drives the coordination process using direct communication. For this purpose, a modified Firefly-based algorithm is proposed as a decision and recruitment mechanism [5].

The above mentioned theoretic approach has been tested in our Java-based simulator, and it has been analyzed in terms of different values of the weights of the objective function, in order to eval-

uate their influence on balancing recruitment and exploration tasks and how they affect the behaviour of the swarm to stick together and/or move according to a common goal. The experiments have been repeated under different conditions such as the variations of the number of targets, the number of required robots to disarm a target, the dimension of the area. In addition, the paper also has explored how the coefficient may influence the performance both in static and dynamic scenarios.

Therefore, the remainder of the paper is organized as follows. Section 2 provides a review of the related work. The descriptions of the problem and the mathematical model are the focus of Section 3 and Section 4. Section 5 describes the proposed bio-inspired approach. Section 6 shows the simulation results obtained from a set of experiments and finally Section 7 outlines the main research conclusions.

2. Related work

2.1. Multi-robot Allocation and Coordination

The applications of multi-objective optimization to evolutionary robotics are diverse with increasing attention. Some common applications for multi-robot teams include foraging [6], path planning [7], search [8], distributed surveillance and security [9], search and rescue [10][11], and emergency service [12, 13, 14]. However, many of these formulations have attempted to solve multiple objectives at the same time and researchers often model these common capabilities by defining them as optimization problems. Typically, these problems can be formulated as combinatorial optimization or as convex optimization problems in special cases so as to take advantage of the many tools available for these type of optimization. However, such formulations were often simplified and were not treated as global optimization problems for multi-robot applications, though the full combinatorial problems related to robot exploration and coordinations can be NP-hard. Since most robotic applications require real-time robot responses, there is insufficient time to calculate globally optimal solutions for most applications; such solutions are generally possible for very small-scale problems. Instead, typical solutions use distributed methods that incorporate only local cost/utility metrics. Although such approaches can only achieve approximations to the global solution, they often can be sufficient for practical applications [15].

On the other hand, a diverse range of studies has been done by imitating ideas from nature for designing control algorithms for multi-robot systems, since the nature presents excellent examples of distributed self-organizing systems. The aim is to develop self adaptive distributed coordination algorithms to deal with the problems. In nature, we often see complex group behaviors arisen from biological systems composed by large numbers of animals that individually lack either the communication and computational capabilities, but they may self-organize, leading to some emerging collective behavior that enable to achieve a common (swarm level) goal [16].

Regarding the exploration task, the main goal is to cover the whole area in the minimum possible time. Therefore, it is essential that the robots are deployed in different regions of the area, at the same time. Most researches have investigated the use of indirect communication as a mechanism to guide the swarm of the robots. Stigmergy is a kind of mechanism that mediates animal-animal interactions through artifacts or via indirect communication, providing a kind of environmental synergy, information gathered from work in progress, distributed incremental learning and memory among the society. Furthermore, pheromone provides a stigmergic medium of communication, which influences the future actions of a single or a group of individuals via the changes made to the environment. Stigmergy allows to record the history of an agents actions without memorizing the information and saving resources [17].

In the exploration task, researchers use the concept of anti-pheromone so as to try to maximize the distance between the robots and to enforce a dispersion mechanism in different sites of the region of interest, with the aim to accomplish the mission as quickly as possible. Some examples of this approach can be found in [18, 19] for surveillance mission, in [20] for guiding the robots in search and rescue in a disaster site, in [21, 22] in multi-robot coverage. Ravankar et al. [23] used a hybrid communication framework that incorporates the repelling behaviour of the anti-pheromone and attractive behaviour of pheromone for efficient map exploration.

Regarding the coordination of the robot for task assignment or allocation, many approaches have been proposed for solving the multi-robot coordination problem. One of the most commonly used swarm-based approaches is the response threshold, where each robot has a stimulus associated with

each task it has to execute. Some response threshold systems use such stimuli and the threshold value for calculating the probability of executing a task [24, 25]. In recent years, market-based approaches have become popular to coordinate multi-robot systems. These methods have attempted to present a distributed solution for the task allocation problem [26]. Jones et al. [27] described a market based approach to task allocation for the fire fighting in a disaster response domain.

Recently, bio-inspired algorithms inspired by a variety of biological systems, have been proposed for self-organized robots. A well known bio-inspired approach takes inspiration from the behaviour of the birds, called Particle Swarm Optimization (PSO). PSO-inspired methods and their extended versions have received much attention and have been applied for the coordination of mobile robots. Some examples of its application can be found for guiding robots for targets searching in complex and noisy environment as presented in [28]. Modified versions of the PSO are proposed to balance searching and selecting in a collective clean-up task [29] for path planning in a clutter environment [30] and for mimicking natural selection emulated using the principles of social exclusion and inclusion [31].

Other studies took inspiration from the bees and ants that mimic the food foraging behaviour of swarms of honey bees and ants in nature. These algorithms have also been applied to robotic systems such as task allocation [32], finding targets and avoiding obstacles [33], for solving on line path planning [34] [35], decision-making to aggregate robots around a zone [13] [16] [36]. A Hybrid approach can be found in [37]. However, extensive reviews of research related to the bio-inspired techniques and the most behaviour of the robots can be found in [38] [39].

2.2. Overview of the Present Paper

As discussed above, the problem of multi-robot coordination has received significant attention. However, the proper coordination for exploration and management of found targets have not been studied extensively. Therefore, the research we report here makes some new contributions in this area:

- The exploration and handling problem is described using an optimization model. Both the search and handling of the hazard targets are

considered together, allowing the trade-off between two aims in terms of a weighted objective function.

- A repulsive mechanism based on ant colony optimization is applied as an indirect coordination mechanism for the exploration task.
- The recruitment task that allows the disarming of a hazardous target cooperatively is modelled using a new bio-inspired Firefly Algorithm based approach to recruit and then coordinate robots movements as a decision mechanism of the robots.
- The environment is highly uncertain where no prior information is available and dynamic events such as robots failures under a limited amount of battery power and unpredictably events in the area can occur.
- The robots react to events that occur in the environment, so unlike other common approaches, they could change, at each time step, the previous decisions taken earlier.

With these more realistic considerations, we present a hybrid bio-inspired approach that allows robots to potentially balance the needs of search and task response in an adaptive way to complete the mission, obtaining full benefits in terms of reducing wastage of resources of the system such as time, energy and mobile alive robots.

3. Scenario and Model Formulation

In this paper we address challenges in twofold: the ability to explore the area to find unknown targets and the ability to efficiently balance multiple (often conflicting) goals. More specifically, a team of robots operate in an unknown environment searching for unknown hazardous targets in order to disarm them cooperatively. The locations of these unknown targets are detected gradually through searching by the robots. A target must be handled by a coalition of robots; therefore, the recruitment task starts in real time as the targets are found.

Once some of these targets are detected, a fixed number of the robots is needed to disarm the target through a predefined procedure of actions, while the remaining robots continue to explore the environment for searching other targets. Assuming that the robots make their decisions independently and

that each of them, in a restricted region of the area, owns the same amount of information, each robot will decide in an autonomous manner which of the two tasks to perform (explore or help the others in disarming process) and in what direction it may move. However, the search and coordination tasks are not entirely decoupled; it is possible for a robot to perform both simultaneously (for example when it moves towards the target, it also implicitly explores the area).

More specifically, the basic structure of our approach is divided into two steps:

1. Detection, evaluation and selection of the cells, while moving during the exploration of the area.
2. Selection among different targets found so far in the area and navigation towards the chosen target.

However the strategy is highly flexible allowing to reconsider, at each time step, the previous decisions taken at an earlier step to react to any new events that may occur.

3.1. Environment modeling

Let A be the 2-D working field or grid, where $A \subset \mathbb{R}^2$. As a symbolic representation, the proposed method uses a grid map with m and n cells. Let us establish a Cartesian coordinate system taking the upper left corner of A as the origin, each cell $c \in A$ of the area has its own definite coordinates (x, y) , with $x \in \{1, 2, \dots, m\}$ and $y \in \{1, 2, \dots, n\}$ elements. The universal set C contains all possible states of a cell on the grid map. The subsets $C_1, C_2, C_3, C_4 \subset C$ (where $C_i \cap C_j = \emptyset, i \neq j$) represent possible states as follows:

- C_1 :{explored by the robots},
- C_2 :{accessible and not explored by the robots},
- C_3 :{occupied by an obstacle},
- C_4 :{not explored and inaccessible after hazardous events (e.g., the mine's explosion) }.

Obstacle cells are inaccessible to the robots and impenetrable to the sensors. A cell occupied, at time step t , by any of the robots is considered as an obstacle, thus no other robot can take the place (see Fig. 1).

The state C_4 is used in the dynamic scenario, as described below, where the found targets such as

Figure 1: A representation of the simulation environment.

(a)(b)

Figure 2: (a) Possible directions of a robot's move (b) Possible angles of a robot's turn.

mines could explode or dangerous chemicals may leak at any time, making the nearby cells inaccessible.

While the robots explore the area, the cells transit to subset C_1 . Each cell $c = (x, y)$ has a maximum of eight adjacent neighbors $N(c)$, if all cells are accessible: $(x - 1, y - 1), (x - 1, y), (x + 1, y), (x, y + 1), (x, y - 1), (x + 1, y + 1), (x - 1, y + 1),$ and $(x + 1, y - 1)$ as shown in Fig. 2.

3.2. Assumptions

A set R of homogeneous robots can be deployed in the area, where $R = \{k \mid k \in \{1, 2, \dots, N^R\}\}$. At each step t , the current state of a robot k can be represented by its coordinates (x_k^t, y_k^t) . The robots are modeled as rational collaborative autonomous agents that move autonomously in the environment. We assume that these robots are identical (executing the same algorithms) and follow simple local rules to communicate with neighbors and with the environment in order to provide a scalable strategy. However, for the sake of the simplicity, the robots are equipped with advanced devices such as sensors, global positioning capabilities (for instance they are equipped with a Wi-fi module) camera, radar, and an onboard automatic target recognition system, with which the robots identify the targets such as mines and obstacles or other robots in proximity. Sensor's information is assumed to be perfect, and we assume that the robots have perfect knowledge of their locations expressed in terms of their coordinates.

They are able to communicate with others using wireless communication and the communication range R_t is limited, compared to the size of area, so two robots can exchange information only if they are close enough; i.e., the distance between them is smaller than R_t . We define a local neighborhood of robot k at time t , denoted by LN_k^t , as the set of robots that are within the R_t of the robot k (Fig. 3).

In addition, it is also assumed that the communication network is perfect (no packets loss, no transmission time or delay), so robots within the

Figure 3: The robots in the cells with coordinates (4,5) and (11,9) have detected a target. They start a recruitment process by sending packets that will be received by the robots within their wireless range R_t .

Figure 4: Local coalition of robots formed through the recruitment process

same wireless range have identical information at the same time. For the ease of presentation, the robots start the search simultaneously at the same time. These assumptions, that can easily be removed in the future, but for the moment this is to simplify the model since we are more interested in analyzing the trade-off in the proposed model.

The robots must explore the area for searching and dealing with a set T of N^T targets such as mines disseminated in the area, i.e., $T = \{z \mid z \in \{1, 2, \dots, N^T\}\}$. It is assumed that there is no prior knowledge about these targets such as the total number and locations. Thus the targets can be located in any position of the area with the same probability. Therefore the only way to ensure the detection of all targets (mines) is to do a complete search of the area.

Each target, is represented by its coordinates (x_z, y_z) . A target z is detected by a robot k when the target's coordinates coincides with the robot's coordinates. Once a robot finds a target, it starts a recruitment process since a target requires a fixed number of robots to be handled. We define R_{min} a non-negative integer represents the number of robots needed to treat safely a target. For this purpose, it exchanges wireless messages locally by sending out help requests through packets (that contains mainly the coordinates of the found target) to the robots within its wireless range R_t . We denote RR_k as a set that keeps track of the help requests that the robot k receives from the others, expressed in terms of found targets, thus $RR_k \subset T$.

Figure 4 shows an example of local coalitions that are formed through the recruitment's processes. Since the robot's decisions can dynamically be changed in terms of robot's movements, new requests, failures, etc. the final configurations in the target's locations can change.

3.2.1. The actions of robots

The actions of robots belong to three main classes:

- I. Sensing actions that affect changes in a robot's knowledge of the environment.
- II. Moving actions in the cells which imply rotation to choose the right directions and obstacles avoidance.
- III. Communication actions when targets are found.

Each robot adapts its position in three different ways. The first is in the direction of minimum amount of pheromone (to indicate good feasible regions unexplored). The second is to move away from other robots or obstacles (to avoid collision). The third is in the direction of the detected targets (to perform an disarming task cooperatively). The first two are based on interactions accumulated over time between the robots and the environment. The third is a reactive behavior triggered by help requests from other robots. In the following, these behavior characteristics are described.

To behave as a collective robotic organism, the robots need to be able to achieve different behavioral states. They are able to reconfigure themselves so as to achieve a transition between the states. More specifically, at the beginning, when no target is detected, each robot collects information from its immediate surrounding cells perceiving chemical substance (pheromone) by on-board sensors and uses this information to identify the direction where to move. Each robot calculates its best move as the next position locally according to an Ant Colony approach as explained below. The goal is that the robots should explore the undetected sub-area as much as possible in order to speed up the task. This state is named the Forager State and it is the initial state for each robot.

Once a robot discovers a target by itself, it will switch to a Coordinator State. Each coordinator robot is responsible for handling the disarmament process of the discovered target and for the recruitment of the others. The recruiting process starts by broadcasting packets to the robots in its wireless range (see Fig. 3), and it ends when the predefined number of necessary robots (R_{min}) arrived to the target's location to form a coalition team. Then, the accumulated robots work together as a group, performing the disarmament task. Essentially a coordinator robot performs the following steps:

1. Check if there are a sufficient number of robots to form a coalition to handle a target.

(a)(b)

Figure 5: (a) The robot in the cell (6,11) that is recruited by the robot in cell (6,8) moves into a cell that is too far from the target, thus it changes its state by becoming an explorer. (b) The robot in cell (9,9) that is recruited by two robots in both cells (6,8) and (10,6), respectively. After, it moves into the cell that is too far from both targets, thus it changes its state by becoming a forager.

2. If there is no coalition that satisfies the constraint, then continue to send packets.
3. Repeat step 2 until all necessary robots are arrived.
4. Otherwise, stop the communication and start to disarm the target properly.

Once the target is disarmed, the involved robots return to continue to explore the area.

When a robot k receives one or more request packets by coordinator robots, it will make the decision whether to continue to explore or to help in a disarming process. If it decides to help a coordinator robot, it forms a list RR_k of the received packets in terms of targets; otherwise, the task is rejected and the robot continues to explore individually. If there is only one request, it has to decide whether it should move to remove the target or continue to explore the area. If more requests have been received, it not only needs to decide whether it should remove, but also it has to decide towards which target it could move. This happens because a robot has to balance the two tasks according to a weighted objective function express below. This state is named the Recruited State.

A key aspect of this state is that the robots react to events that occur. Unlike common approaches, they could change the decisions taken previously during the iterations. For example, for a certain type of mission, it is possible to meet a target or receive different requests, while reaching another target in response to a recruitment process, thus reconsidering the choice of the target to be handled. Moreover, the decision can be to restart to explore the area since the movements are too far from the target's location see Fig. 5.

When a recruited robot, once it reaches the target's location, it will wait until the other needed robots to arrive and thus enter into the waiting mode. This state is called the Waiting State.

Figure 6: State transition logic for robots at each time step. `WithinRange()` is a function returning `True` when the robot is within the communication range of a coordinator robot. `Confirm Recruiting()` is a function returning `True` if the robot decides to get involved in the disarmament process of a target. `ArrivedToMineLocation()` and `ReceivedInstructionsCoordinator()` are two functions returning `True` if the robot is arrived to a target's location and received the command to start the disarmament process by the coordinator, respectively. `NecessaryRobot()` returns `True` if all needed robots (R_{min}) have arrived at the target's location.

Finally, once the number of needed robots reach the target's location, the group as a whole is involved in the disarming process and they will perform, for a fixed amount of time, some actions to deal with the targets properly. This state is the Execution State.

To summarize the above actions and states, Fig. 6 shows the state transition logic of robots at each time step.

4. Multi-Objective Optimization Formulation

Multiple conflicting objectives may arise naturally in most real-world robotic optimization problems. Several principles and strategies have been developed and proposed for over the last decades in order to solve such problems. In multi-objective optimization, as its name implies, there are multiple objective functions with a possibility of conflicting with each other. The aim is to find a set of vectors of decision variables that can satisfy constraints and optimize (minimizes or maximizes) these functions. Such solution vectors are not a unique vector, there are many such solutions vectors forming a so-called Pareto front. Each point or non-dominated solution on the Pareto front provides a preference and choice between different objectives. When the Pareto front becomes convex, weighted sum methods can aggregate different objectives into a single objective.

In general, a multi-objective optimization problem can be written mathematically as

$$\begin{aligned}
 &\text{To find vectors } X = (x_1, x_2, \dots, x_L)^T \in \Omega \\
 &\text{which optimize } f(x) = (f_1(x), f_2(x), \dots, f_p(x)) \\
 &\text{subject to } g_j(x) \leq 0 \quad j = 1, \dots, s, \\
 &\quad h_r(x) = 0, \quad r = 1, \dots, d, \quad (1)
 \end{aligned}$$

where $f_1(x), f_2(x), \dots, f_p(x)$ denote the objective functions to be optimized simultaneously, X

is the vector of the decision variables in the search/decision space. Ω is the set of feasible solutions and $g_j(x)$ denotes the inequality constraints, while $h_r(x)$ are equality constraints. All these functions can be linear or nonlinear [40].

As it is very difficult to effectively handle with all the conflicting objective functions, several methods have been developed for this purpose. One of these methods is that the multi-objective problem is transformed into a single-objective problem by a weighted sum. In this paper, in order to solve our bi-objective problem, the weighted sum method is used to deal with conflicting goals and the solutions can be obtained as a trade-off of the specific problem. The total cost of the fitness function is obtained by a linear combination of the weighted sum of two objectives in which each objective function based on its importance or preference [41].

The problem is transformed into a single-objective optimization problem by using scalar weighting factors associated with each objective function as follows:

$$F_{\text{weighted sum}} = \sum_{i=1}^p w_i f_i(x) \quad (2)$$

where w_i

$$w_1 + w_2 + \dots + w_p = 1, \quad w_i \geq 0. \quad (3)$$

The weighted sum method changes weights systematically, and each different single objective optimization determines a different optimal solution. This approach gives an idea about the shape of the Pareto front and allows information to be obtained about the trade-off among the various objectives to accumulate gradually [42].

4.1. Formulation as a Bi-objective Problem

In order to define the problem as a bi-objective optimization model, there are different decisions to be made by a robot. Given the position, expressed by the coordinates, where each robot (e.g., robot k) is located at each time step, and given a found target $z \in RR_k$, then the robot k has to decide if it will get involved in the recruiting process or will continue to explore the area.

4.1.1. First Objective

Firstly, the first decision can mathematically be represented as the following decision variable:

$$v_{xy}^k = \begin{cases} 1 & \text{if the robot } k \text{ visits the cell } (x, y), \\ 0 & \text{otherwise.} \end{cases}$$

(4)

We assume that the time to visit a cell is denoted by T_e and it is supposed to be the same for all robots. Then the goal of an exploration task is to cover the whole area in the minimum amount of time, and thus the first objective becomes

$$\text{minimize } \sum_{k=1}^{N^R} \sum_{x=1}^m \sum_{y=1}^n T_e v_{xy}^k. \quad (5)$$

4.1.2. Second Objective

Similarly, the following decision variable allows us to model if a robot k is involved in the disarmament process of a found target z :

$$u_z^k = \begin{cases} 1 & \text{if robot } k \text{ is involved with target } z, \\ 0 & \text{otherwise.} \end{cases} \quad (6)$$

When a robot has eventually detected a target, it should act as an attractor, trying to recruit the required number of robots so as to disarm the discovered target safely and properly. As mentioned above, it is necessary to form a coalition to handle a target. We define the number of robots that can deal with a target as R_{min} . Therefore, it sends help requests using wireless communication via packets that are received by the other robots in the wireless range. It should be noted that the received packets are not forwarded, since we are focused on one hop communication. Then, the decision is taken by the recruited robots. The quality of the recruitment process is measured in terms of time.

Let $T_{Start,z}^k$ be the time step at which the robot k receive a help request for disarming the target z and $T_{End,z}^k$ the time step at which the robot k has reached the target z , then $(T_{End,z}^k - T_{Start,z}^k)$ is the time taken for coordination, namely, the coordination time. Thus, the objective is the minimization of the coordination time for each found target, in order to speed up the disarming process and continue the mission effectively. Therefore, the second objective is

$$\text{minimize } \sum_{k=1}^{N^R} \sum_{z=1}^{N^T} (T_{End,z}^k - T_{Start,z}^k) u_z^k. \quad (7)$$

4.1.3. The Bi-Objective Optimization Problem

The considered objective function is thus related to the minimization of the time needed to perform

the overall mission. The problem of selecting the best solutions for the problem (in the Pareto sense), accounting both the exploration time and the coordination time, can be mathematically stated as follows:

$$\begin{aligned} \text{minimize } & \sum_{k=1}^{N^R} \sum_{x=1}^m \sum_{y=1}^n T_e v_{xy}^k, \quad \text{and} \\ \text{minimize } & \sum_{k=1}^{N^R} \sum_{z=1}^{N^T} (T_{End,z}^k - T_{Start,z}^k) u_z^k, \end{aligned} \quad (8)$$

subject to

$$\sum_{k=1}^{N^R} v_{xy}^k \geq 1, \quad \forall (x, y) \in A, \quad (9)$$

$$\sum_{k=1}^{N^R} u_z^k = R_{min}, \quad \forall z \in T, \quad (10)$$

$$v_{xy}^k \in \{0, 1\}, \quad \forall (x, y) \in A, k \in R, \quad (11)$$

$$u_z^k \in \{0, 1\}, \quad \forall z \in A, k \in R, \quad (12)$$

$$T_e, T_{End,z}^k, T_{Start,z}^k \in R, \quad \forall z \in A, k \in R. \quad (13)$$

The objective functions in (8) to be minimized represent the total time consumed by the swarm of robots. They depend on both the time for the exploration of the area and the time for coordinating the robots involved in the disarming process of the targets. Constraint (9) ensures that each cell is visited at least once. Constraint (10) defines that each target z must be disarmed safely by R_{min} robots. The constraints (11)-(13) specify the domain of the decision variables.

The problem was formulated as a bi-criteria model which turns out to be very challenging to solve. Indeed, the number of efficient solutions may be exponential in terms of the problem size, thus prohibiting any efficient method to determine all efficient solutions. For these reasons, following the popular approaches used to deal with multi-objective optimization problems, the model has been transformed into a single objective optimization problem using arbitrary importance factors for each criteria (i.e. w_1 and w_2) and combining the objectives as a single function to be minimized.

The resulting single objective problem with non-negative weights can be represented as follows:

$$\begin{aligned} \text{minimize } & \sum_{k=1}^{N^R} \sum_{x=1}^m \sum_{y=1}^n w_1 (T_e v_{xy}^k) \\ & + \sum_{k=1}^{N^R} \sum_{z=1}^{N^T} w_2 [(T_{End,z}^k - T_{Start,z}^k)] u_z^k, \end{aligned} \quad (14)$$

subject to constraints (9)-(13).

Parameters w_1 and w_2 are chosen such that the condition $w_1 + w_2 = 1$ is satisfied. In this case, the combined function is Pareto optimal [42]. The user can choose appropriate values for the parameters w_1 and w_2 , depending on the preference or priority of the objectives. Indeed, by minimizing the weighted sum objective with various settings, it is possible to determine various points in the Pareto set. This approach can approximate and describe the shape of the Pareto front effectively, allowing the accumulation of information to be obtained about the trade-off among various objectives.

The proposed single objective optimization model can be solved and be relevant to many applications for robot exploration and coordination. For applications in which more relevance is given to the exploration task, more importance could be given to exploration time (thus higher value of w_1), whereas for applications where it is more important to reduce the disarming time, more importance could be given to w_2 .

By minimizing the overall fitness function in regard to the assigned weights of each criterion, a suitable decision mechanism that may balance the two objectives can be obtained. The weights have been tuned through a set of simulations in order to try to find the best values.

4.2. Energy Model

For each activity executed by a robot k , a certain amount of energy is consumed. In our study, the energy model reflects mostly two activities: energy for communication and energy for mobility. The mobility energy depends on several factors. For simplicity, the mobility cost for a robot k in our model can be calculated by considering the distance traversed in terms of the number of cells and it is expressed as follows:

$$E_{mob}^k = \sum_{x=1}^m \sum_{y=1}^n C_{mob} v_{xy}^k, \quad (15)$$

where $\sum_{x=1}^m \sum_{y=1}^n v_{xy}^k$ is the total number of visited cells for each robot k while moving in the exploration phase and recruiting phase; C_{mob} is the energy cost given to move from one cell to another and takes into account both the costs for moving and turning [43].

When a target is detected, the energy consumed is instead related to the communication and to the cost for performing the planned task on the target. Since we use a wireless communication system in this phase, the energy consumed depends on the transmission and reception of the packets to communicate the position of the targets. In this case, we assume that the energy consumed by robot k to transmit E_{tx}^k and receive E_{rx}^k a packet [44] is related to the maximum transmission range R_t and to the packet size (l) as follows:

$$E_{tx}^k = l (R_t^\psi e_{tx} + e_{cct}), \quad (16)$$

where e_{tx} is the energy required by the power amplifier of transceiver to transmit one bit data over the distance of one meter, and e_{cct} is the energy consumed in the electronic circuits of the transceiver to transmit or receive one bit. Here, ψ is called the path loss exponent of the transmission medium where $\psi \in [2, 6]$.

On the other hand, the energy consumption for receiving a packet is independent of the distance between communication nodes and it is defined as:

$$E_{rx}^k = l e_{rc}, \quad (17)$$

The energy consumed to deal with a target is:

$$E_d^k = C_d, \quad (18)$$

where C_d is the cost given to the working task for handling a target properly, and it is the same for each robot and it is related, for simplicity, to the mechanical movements. Essentially, we model the energy consumed for the coordination task by the robot k that is involved in the targets issue as:

$$E_{coord}^k = \sum_{z=1}^{N^T} (E_{tx}^k + E_{rx}^k + E_d^k) u_z^k. \quad (19)$$

Based on the previous considerations and models, we now introduce a performance index, called Total-Energy-Swarm-Consumption (TESC), as:

$$TESC = \sum_{k=1}^{N^R} E_{mob}^k + \sum_{k=1}^{N^R} E_{coord}^k. \quad (20)$$

That is, the total energy consumed by the swarm of robots and it is the sum of two contributions: energy consumption for moving into the area and energy consumption for the wireless communication when they are involved in the performing of the targets.

4.3. Robot in dynamic scenarios

The above considerations provide a unified approach to consider both the complete discovery of the area and the measure of the time needed to accomplish both exploration and disarming of the targets. Thus, it is a useful metric, but it requires that the task is completely finished, and cannot be used to evaluate partial execution of the tasks. In many cases a complete exploration of the environment may not be feasible in practice, due to the time or resource constraints in large and hazardous environments.

In this section, we consider a dynamic environment in the sense that the targets can explode at any time and in an unpredictable manner, mimicking the destruction of some robots and the damage of the nearby zones. Moreover, we regard the robots with a limited quantity of energy without the possibility of recharge or replacement. In such scenarios, the team works under more demanding time constraints.

In order to use a performance metric that is applicable to the robotic system in a dynamic scenario, we have considered several functions; one for each feature that must be discovered and measured from the environments. More specifically, the performance metrics are given by each function measuring the percentage/ratio of information related to the two tasks. In the case of exploration task, it is the percentage of the environment explored not covered by impassable obstacles, while in the case of the disarmament task, it is the percentage of targets successfully identified and disarmed.

The following equations summarize the region of an emergency scene as follows:

$$A_E = \sum_{x=1}^m \sum_{y=1}^n c_{(xy)} \quad c_{xy} \in C_1, \quad (21)$$

$$A_{UN} = \sum_{x=1}^m \sum_{y=1}^n c_{(xy)} \quad c_{xy} \in C_2, C_4. \quad (22)$$

Concerning the above regions of interest, we define the following terms:

$$F_1 = \frac{A_E}{\sum_{x=1}^m \sum_{y=1}^n c(xy)}, \quad c_{xy} \in C_1, C_2, C_4 \quad (23)$$

where F_1 is a regularized term that indicates the percentage of explored cells in the emergency scene. Thus, F_1 will be equal to one only in the case all cells of the area have been explored, except for the cells with obstacles ($c \in C_3$).

Now we define the number F_2 of handled targets as follows:

$$F_2 = \sum_{z=1}^{N^T} f(z) = \begin{cases} 1 & \text{if target } z \text{ is disarmed properly,} \\ 0 & \text{otherwise.} \end{cases} \quad (24)$$

In this case, the objectives essentially become the maximization of the percentage of explored area and the number of disarmed targets. In this case, the robots have a limited amount of energy and at each time step, a fixed quantity of energy is consumed (see Section 4.2) depending on what action the robot may perform and if the mines can explode. The mission can terminate for multiple reasons, including the case that all robots have used up the energy, or are damaged due to explosion.

5. Proposed Bio-Inspired Approaches

Our proposed approach to solve the problem is a hybrid strategy which combines both indirect and direct communication mechanisms. We study how robots can accomplish the mission in a distributed and self-organized way through a stigmergic process in the exploration task, and simple information locally sent by the robots in the recruitment task. Our system has unique features such as the minimal information exchange, and local interactions between simple homogeneous robots, achieving complex collective behaviour. Such solutions are in line with the general approaches used in swarm robotics, and support the desired system properties of robustness, adaptivity and scalability.

5.1. Repulse Pheromone-Based Strategy for Exploration

The mobile, autonomous robots, performing the exploration search task, must be able to decide the sequence of movements needed to explore the whole

environment. In this work, we address the exploration problem in the context of search and rescue operations. Exploration strategies that drive the robots around an unknown environment on the basis of the available knowledge are fundamental for an effective search. The mainstream approaches for developing exploration strategies are mostly based on the idea of incrementally exploring environment by evaluating a number of candidate observation locations, in our cases neighbor cells, according to a criterion and by selecting, at each step the next best location. However, we do not address the problem to build a map of the environment, since we are more interested in locating the largest number of targets in the minimum amount of time. Differently from map building, search and rescue settings are strongly constrained by both time and battery limitations and generally require the amount of explored area over the map quality. Since the robots should be required to be capable of various functionalities other than area exploration, it is desirable that both the integration to a swarm and the ability to explore are seamless and these actions should not consume a large amount of the robots resources.

To be effective, a search strategy must attract robots towards unobserved areas so as to avoid the undesirable scenario where some areas are frequently revisited while others remain unexplored. Therefore, some swarm control is needed. The swarm control algorithm used here is a pheromone-inspired mechanism in which the environment is assumed to handle the storing and diffusion of chemical substance; thus, the robot controllers do not store any chemical information, except the sampled concentrations within the immediate vicinity of the robot. Utilizing a stigmergic communication would be an efficient method of achieving such emergent behaviour with low overhead.

Essentially, when the robots are exploring the area, they lay pheromone on the traversed cells and each robot uses the distribution of pheromone in its immediate vicinity to decide where to move. Like in nature, the pheromone trails change in both space and time. The pheromone deposited by a robot on a cell diffuses outwards cell-by-cell until a certain distance R_s such that $R_s \subset A \subset \mathbb{R}^2$ and the amount of the pheromone decreases as the distance from the robot increases (see Figure 7).

Mathematically, the pheromone diffusion is defined as follows: consider that robot k at iteration t is located in a cell of coordinates $(x_k^t, y_k^t) \in A$, then

Figure 7: Example of pheromone diffusion. When a robot moves into new cell, it spreads the pheromone within a certain distance R_s . The intensity of pheromone decays according to the distance from the cell.

the amount of pheromone that the robot deposits at the cell c of coordinates (x, y) is given by:

$$\Delta\tau_c^{k,t} = \begin{cases} \Delta\tau_0 e^{-\frac{r_{kc}}{a_1}} - \frac{\varepsilon}{a_2} & \text{if } r_{kc} \leq R_s, \\ 0 & \text{otherwise,} \end{cases} \quad (25)$$

where r_{kc} is the distance between the robot k and the cell c and it is defined as:

$$r_{kc} = \sqrt{(x_k^t - x)^2 + (y_k^t - y)^2}. \quad (26)$$

This means that pheromone spreads up to a certain distance, as in the real world, after which it is no perceivable by other robots. In addition, $\Delta\tau_0$ is the quantity of pheromone sprayed in the cell where the robot k is placed and it is the maximum amount of pheromone, ε is an heuristic value (noise) and $\varepsilon \in (0, 1)$. Furthermore, a_1 and a_2 are two constants to reduce or increase the effect of the noise and pheromone. It should be noted that multiple robots can deposit pheromone in the environment at same time, then the total amount of pheromone that can be sensed in a cell c depends on the contribution of many robots.

Furthermore, the deposited pheromone concentration is not fixed and evaporates with the time. The rate of evaporation of pheromone is given by ρ ($0 \leq \rho \leq 1$), and the total amount of pheromone evaporated in the cell c at step t is given by the following function:

$$\xi_c^t = \rho \tau_c^t, \quad (27)$$

where τ_c^t is the total amount of the pheromone on the cell c at iteration t .

Considering the evaporation of the pheromone and the diffusion according to the distance, the total amount of pheromone in the cell c at iteration t is given by

$$\tau_c^t = \tau_c^{(t-1)} - \xi_c^{(t-1)} + \sum_{k=1}^{N^R} \Delta\tau_c^{k,t}. \quad (28)$$

5.2. Cells Selection

At each time step, the algorithm selects the most appropriate cell for each robot, among a set of

neighbor cells without the knowledge of the entire area. This happens because the robots have not got global information about the environment. The aim is to avoid any overlapping and redundancy efforts, therefore, the robots must be highly dispersed in the area in order to complete the mission as quickly as possible, avoiding at the same time any wastage of the robot's resources such as energy.

Each robot k , at each time step t , is placed on a particular cell c_k^t that is surrounded by a set of accessible neighbor cells $N(c_k^t)$. Essentially, each robot perceives the pheromone deposited into the nearby cells, and then it chooses which cell to move to at the next step. The probability at each step t for a robot k of moving from cell c_k^t to cell $c \in N(c_k^t)$ can be calculated by

$$p(c|c_k^t) = \frac{(\tau_c^t)^\varphi (\eta_c^t)^\lambda}{\sum_{b \in N(c_k^t)} (\tau_b^t)^\varphi (\eta_b^t)^\lambda}, \quad \forall c \in N(c_k^t), \quad (29)$$

where $(\tau_c^t)^\varphi$ is the quantity of pheromone in the cell c at iteration t , and $(\eta_c^t)^\lambda$ is the heuristic variable to avoid the robots being trapped in a local minimum. In addition, φ and λ are two constant parameters which balance the weight to be given to pheromone values and heuristic values, respectively. The robot k moves into the cell that satisfies the following condition:

$$c = \arg \min [p(c|c_k^t)]. \quad (30)$$

In this way, the robots will prefer less frequently visited regions and it is more likely that it will direct towards an unexplored region. The exploration strategy is detailed in Algorithm 1 and it was previously validated [45]. At the first iteration of Algorithm 1, all cells are initialized with the same value of the pheromone trail, set to be zero that represents that the cells have not yet been visited by any of the robots, so that the initial probabilities that a cell would be chosen is almost random. Then the robots move from a cell to another based on the cell transition rule in Eq.(29). Unvisited cells become more attractive to the robots in the subsequent iterations. Using this approach, the robots explore the area by following the flow of the minimum pheromone. Then the pheromone trails on the visited cells by ants are updated as in Eq. (28).

Algorithm 1 stops executing for a robot when it becomes a coordinator or it is recruited or if the mission is completed (that is all cells have been

Figure 8: The flow chart of Exploration task for a robot

visited at least once and all targets are found and disarmed), therefore the frequency of its execution depends on the state of the robots during the mission. Fig. 8 illustrates a simplified flowchart of the ACO-based strategy applied by each robot agent in exploration states.

5.3. Recruitment task

This task aims to design a low-cost coordination mechanism that is able to form groups of robots at given sites where the targets are found. Once a robot detects a target (mine), since it does not have sufficient resource capabilities to handle it, it acts as a strong attractor to the other robots within the wireless range to form a coalition that cooperatively works for the disarmament process of the target. The detection of a target may happen at any time during the exploration of the area, so the recruitment process can take place in different regions of the area.

For this purpose, wireless communication is used to share the information about the found targets, since direct communication may be beneficial when a fast reaction is expected and countermeasures must be taken. In this case, each robot is assumed to have transmitters and receivers, using which it can send packets to other robots within its wireless range R_t and there is no propagation of the packets (one hop communication) as shown in Fig. 3. The packets contain mostly the coordinates of the detected targets. Therefore, the volume of information that is communicated among the robots is small, but it implies the robots still lack global knowledge of the environment.

The most common approach is in a greedy fashion in which the target is instantaneously assigned to the robots without taking into account future events. Here, we propose a flexible strategy in which the robots not only balance the two tasks such as exploration and manipulation of the targets (mine), but can react to future new events changing, eventually, the taken decisions. However, each robot must take individual decisions that could lead to retract itself from help requests. For example, for such kind of mission, it is possible to detect a target, while reaching another, or to receive another request, and thus may change decisions to move in a more convenient way from the robot's point of

Algorithm 1: Exploration algorithm inspired by ant colony optimization.

```

begin
  Step 1: Initialization.
  Set  $t$ : { $t$  is the time step}.
  Define  $\varphi, \lambda, a_1, a_2, \epsilon, \Delta\tau_0,$ 
   $\rho, R_s$ 
  Step 2: Generation coordination system.
  For the whole swarm, set the
  initial locations in terms of
  coordinates in  $x$  and  $y$  directions.
  Step 3: Procedure
  while the stop criteria are not satisfied do
    foreach robot  $k$  in Forager State do
      evaluate the current position  $c_k^t$ ;
      evaluate neighborhood  $N(c_k^t)$ ;
      compute  $c$  according Eq. (30);
      if ( $c.hasObstacle()$  or
       $c.isOccupated()$  or
       $c.isInaccessible()$ ) then
        choose a random cell  $c^* \in N(c_k^t)$ ;
        move robot  $k$  towards  $c^*$ ;
      else
        move robot  $k$  towards  $c$ ;
        deposit pheromone according to
        Eq. 25;
      end if
    end foreach
    foreach cell  $c \in A$  do
      update pheromone according
      Eq.(28);
    end foreach
    update  $t$ ;
  end while
end

```

view. So at each time step, the robots will make the best selfish decision based on their positions, in response to the received help requests, trying at the same time to balance the two tasks.

It is worth mentioning that the decisions to be made by the robots are independent, and the other robots and the coordinators do not know the taken decisions; therefore, the coordinators robots will continue to send packets until the needed robots have actually arrived.

5.3.1. Firefly-based Decision Mechanism for the Cooperation Task

Firefly Algorithm (FA) is a nature-inspired stochastic global optimization method that was developed by Yang [5]. It tries to mimic the flashing behaviour of a swarm of fireflies. A firefly in the search space communicates with the neighboring fireflies through its brightness which influences the selection.

Fireflies swarm in nature exhibit social behaviour that use collective intelligence to perform their essential activities like species recognition, foraging, defensive mechanism and mating. A firefly has a special mode of communication with its light intensity that signals to the swarm about its information concerning its species, location, attractiveness and so on. The two important properties of the firefly's flashing light are defined as follows:

- brightness of the firefly is proportional to its attractiveness, and
- brightness and attractiveness of pair of fireflies is inversely proportional to the distance between two.

These properties are responsible for the visibility of fireflies which pave way to communicate with each other.

In the proposed approach concerning our robot problem, each coordinator robot that has found a target starts to behave like a firefly. At each time step, the probability of selection of a firefly is higher when the intensity value of the flashes is high. The intensity value depends on the distance. So the recruited robot moves towards the most attractiveness firefly/robot in the range. Attractiveness decreases as the distance increases. The distance r_{ij} between any two fireflies i and j , at positions x_i and x_j , respectively, can be defined as the Euclidean distance as follows:

$$r_{ij} = \|x_i - x_j\| = \sqrt{\sum_{d=1}^D (x_{i,d} - x_{j,d})^2}, \quad (31)$$

where $x_{i,d}$ is the d th component of the spatial coordinate x_i of the i th firefly and D is the number of dimensions. In 2-D case, $r(i, j) : \mathbb{R}^2 \rightarrow \mathbb{R}$

$$r_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}. \quad (32)$$

In the firefly algorithm, as the attractiveness function of a firefly j varies with distance, one

should select any monotonically decreasing function of the distance to the chosen firefly. For example, we can use the following exponential function:

$$\beta = \beta_0 e^{-\gamma r_{ij}^2}, \quad (33)$$

where r_{ij} is the distance defined as in Eq. (31), β_0 is the initial attractiveness at the distance $r_{ij} = 0$, and γ is an absorption coefficient at the source which controls the decrease of the light intensity. The movement of a firefly i which is attracted by a more attractive (i.e., brighter) firefly j is governed by the following evolution equation:

$$x_i^{t+1} = x_i^t + \beta_0 e^{-\gamma r_{ij}^2} (x_j^t - x_i^t) + \alpha(\sigma - \frac{1}{2}), \quad (34)$$

where the first term on the right-hand side is the current position of the firefly i , the second term is used for modelling the attractiveness of the firefly as the light intensity seen by adjacent fireflies, and the third term is randomization with α being the randomization parameter and it is determined by the problem of interest. Though σ was originally a random variable, we use it here as a scaling factor that controls the distance of visibility and in most case we can use $\sigma \in [0, 1]$.

Broadly speaking, when a robot k detects a target, it switches to a Coordinator State, and it acts as a firefly, sending out help requests to its neighborhood LN_k^t . When a robot receives this request and it decides to contribute in the disarming process, it stores the request in its list RR_K . If the list contains more requests, it must choose which target it will disarm. Using the relative position information of the found targets, the robot derives the distance between it and the coordinators and then uses this metric to choose the best target, that is usually the closer. The same information also allows to derive the next movement of the robots. The approach provides a flexible way to decide when it is necessary to reconsider decisions and how to choose among different targets.

It should be noticed that the recruited robots do not respond to the received requests, since they can change their decision at any time, so the coordinators robots do not know which robots are recruiting and continue to broadcast packets until the needed robots have arrived. This has some implications. First, not all recruited robots will go towards the target's locations balancing the two task. Second, the order on which the requests are received is not as important as the allocation is not instantaneous.

Figure 9: Example of an overlap region in which some robots are in the wireless ranges of different coordinator robots and thus they must decide towards which target to move, according to the Firefly-based strategy.

This allows an effective approach to reach solutions that the greedy strategy would miss. Third, the reduction of the impact on communications, so that bandwidth used will increase slowly with the team size.

Then the robots move towards target's location according to a modified version of the firefly algorithm. The aim of this strategy is to increase the flexibility of the system that let the robots be able to form groups effectively and efficiently in order to enhance the parallelism of the handling of the found targets, and at the same time move towards the target location's avoiding overlapping regions and any redundancy (Fig. 9).

Moreover, the algorithm allows to dynamically adjust the coordination task since it allows for each robot to make the best choice from its own point of view.

5.3.2. Implementation of Robot Decision Mechanism

The original version of FA is applied in the continuous space, and cannot be applied directly to tackle discrete problems, so we have modified the algorithm in order to solve our problem. In our case, a robot can move in a 2-D discrete space and it can go just to the adjacent cells. This means that when a robot k , at iteration t , in the cell c_k^t with coordinates (x_k^t, y_k^t) receives a packet by a coordinator robot that has found a target, this robot will move in the next step ($t + 1$) to a new position (x_k^{t+1}, y_k^{t+1}) , according to the FA attraction rules such as expressed below:

$$\begin{cases} x_k^{t+1} = x_k^t + \beta_0 e^{-\gamma r_{kz}^2} (x_z - x_k^t) + \alpha(\sigma - \frac{1}{2}), \\ y_k^{t+1} = y_k^t + \beta_0 e^{-\gamma r_{kz}^2} (y_z - y_k^t) + \alpha(\sigma - \frac{1}{2}), \end{cases} \quad (35)$$

where x_z and y_z represent the coordinates of the selected target translated in terms of row and column of the matrix area, r_{kz} is the Euclidean distance between the target z and the recruited robot. It should be noticed that the targets are static and a robot can receive more than one request. In the latter case, it will choose to move towards the brighter

target within the minimum distance from the target as expressed in Eq. (33). A robot's movement is conditioned by the target's position and by a random component that it is useful to avoid the situation that more recruited robots go towards the same target if more targets have found. This last condition enables to the algorithm to potentially jump out of any local optimum (Fig. 9).

A key aspect occurs when robot k , moves too far from the target's position. Given a robot k located at the step t in the cell of coordinates (x_k^t, y_k^t) and the target z with coordinates (x_z, y_z) , we define the distance between the robot k and the target z as the Euclidean distance

$$r_{kz} = \sqrt{(x_k^t - x_z)^2 + (y_k^t - y_z)^2}.$$

If $r_{kz} \geq (R_t + \Delta) \forall z \in RR_k$ means that the robot k moves too far from the target's locations and in this case, if it has not got other requests, it switches its role into Forager State (see Fig. 5).

Fig. 10 summarizes the main idea of the FA-based decision mechanism.

In order to modify the FA to a discrete version, the robot movements have been modelled by three kinds of possible value updates for coordinates $\{-1, 0, 1\}$, according to the following conditions:

$$\begin{cases} x_k^{t+1} = x_k^t + 1 & \text{if } [\beta_0 e^{-\gamma r_{kz}^2} (x_z - x_k^t) + \alpha(\sigma - \frac{1}{2}) > 0], \\ x_k^{t+1} = x_k^t - 1 & \text{if } [\beta_0 e^{-\gamma r_{kz}^2} (x_z - x_k^t) + \alpha(\sigma - \frac{1}{2}) < 0], \\ x_k^{t+1} = x_k^t & \text{if } [\beta_0 e^{-\gamma r_{kz}^2} (x_z - x_k^t) + \alpha(\sigma - \frac{1}{2}) = 0], \end{cases} \quad (36)$$

and

$$\begin{cases} y_k^{t+1} = y_k^t + 1 & \text{if } [\beta_0 e^{-\gamma r_{kz}^2} (y_z - y_k^t) + \alpha(\sigma - \frac{1}{2}) > 0], \\ y_k^{t+1} = y_k^t - 1 & \text{if } [\beta_0 e^{-\gamma r_{kz}^2} (y_z - y_k^t) + \alpha(\sigma - \frac{1}{2}) < 0], \\ y_k^{t+1} = y_k^t & \text{if } [\beta_0 e^{-\gamma r_{kz}^2} (y_z - y_k^t) + \alpha(\sigma - \frac{1}{2}) = 0]. \end{cases} \quad (37)$$

A robot (e.g., robot k) that is in the cell with coordinates (x_k^t, y_k^t) as depicted in Fig. 11 can move into eight possible cells according to the three possible values attributed to x_k and y_k . For example, if the result of Eqs. (36)-(37) is $(-1, 1)$, the robot

Algorithm 2: Firefly based strategy.

```
begin
  Step 1: Initialization. Set  $t$  { $t$  is the
    time step};
    Set the detected targets;
    Set the robots in Recruited State;
    Define the light absorption
    coefficient  $\gamma$ ;
    Set the randomization parameter
     $\alpha$ ;
    Set the random number  $\sigma$ ;
    Set the attractiveness  $\beta_0$ ;
  Step 2: Generation coordination
    system.
    For the detected targets and the
    recruited robots, set the initial
    locations in terms of coordinates
    in  $x$  and  $y$  directions;
  Step 3: Procedure.
  while The stop criteria are not satisfied do
    foreach robot  $k$  in Recruited State do
      set  $RR_k$ ;
      evaluate the current position  $c_k^t$ ;
      foreach target  $z \in RR_k$  do
        evaluate  $\beta$  according to Eq.
        (33);
        choose the best target  $z$ ;
      end foreach
      evaluate  $N(c_k^t)$ ;
      compute the cell  $c_k^{t+1}$  according to
      Eqs.(36)-(37);
      if ( $c_k^{t+1}.hasObstacle()$  or
         $c_k^{t+1}.isOccupated()$  or
         $c_k^{t+1}.isInaccessible()$ ) then
        choose a random cell  $c^* \in N(c_k^t)$ ;
        move robot  $k$  towards  $c^*$ ;
      else
        move robot  $k$  towards  $c_k^{t+1}$ ;
      end if
    end foreach
    update  $t$ ;
  end while
end
```

will move into the cell $(x_k^t - 1, y_k^t + 1)$. In the described problem, the algorithm for the Firefly based strategy is shown in Algorithm 2.

The Algorithm 2 is executed when one or more targets are found and some robots are recruited by others. If no target are detected or all targets are

Figure 10: The flow chart of the Firefly Algorithm executed by a robot. At each step, the robots make decisions on the basis of events that can occur.

Figure 11: A possible selected cell after the application of a bio-inspired strategy.

removed or handled, the robots perform the exploration task according to Algorithm 1. More specifically, each recruited robot has the list of the requests in terms of target's locations and evaluates the brightness of each of them encoded as fireflies taking into account their distances. At each step, the robots select the best from their list which has the maximum brightness. Next they move to the target's location according to Firefly-based rules.

The proposed firefly-based approach is computationally simple. It requires only few simple calculations (e.g., additions/subtractions) to update the positions of the robots. Moreover, the volume of information that is communicated among the robots is small, since only the position of the targets's is sent. For this reason, FTS-RR has the benefit of the scalability. In addition, the algorithm tries to form a coalition with the minimum size of involved robots, so the remaining robots are able to conduct other search or disarmament tasks, allowing multiple actions at a time.

5.4. Advantages

A detailed analysis of the above proposed approach, we can highlight that this hybrid approach combining indirect and direct communication has the following advantages:

- It allows to dynamically adjust the importance of the various objectives used in order to get the best taken decision during each algorithm iteration.
- It introduces the concept of the importance of the task in terms of the weighted objective function, enabling robots to balance the two objectives when necessary.
- One of the most significant advantages of the approach is the flexibility of change decision at any time.
- It has a low computation cost since it is not required to know the decisions of the other

robots, but each robot acts selfishly taking the best decision from its own point of view.

- Since the algorithm for the coordination is not constructed for the specific target type, all of them are treated in the exact same manner: the same variable types are used, regardless of what the target is and what the robot is performing. Therefore, the approach is generalized and can be used for a wide range of applications with minor modifications.
- Although each independent task is executed individually, the whole system can attempt to globally optimize the process.
- The system can adapt to changes in the environment, and thus the algorithm is scalable.

6. Computational Experiments

A set of experiments has been performed in order to show and analyze the effects of the weights on the performance of the swarm of robots. Simulation results have been summarized using different values of w_1 and w_2 in order to find the best configurations to balance the two tasks. More specifically, we have evaluated the values of the weights under different conditions and by varying the parameters of the system. A simulator has been implemented in Java and for each set of experiments, the simulations have been repeated 50 times, thus the presented results are the mean values of those iterations. Each experiment starts with a random configuration of robots and targets.

6.1. Test Parameters and Metrics

An important aspect in evaluating search and rescue tasks is the definition of effective metrics for measuring the performance of the swarm. The performance metrics, in this paper, can be divided into two groups: measure of the time or energy needed to acquire all the information of the environments and the measure of information acquired with limited resources in terms of time, energy or other constraints.

The first case considers a static scenario where it is supposed that the robots have enough resources to explore the area and disarm all disseminated targets. To measure the performance, we have used two metrics that are the total time steps to complete the mission and the total energy consumed

Table 1: Parameters used in the exploration algorithm.

Parameters	Value
Sensing range R_s	4
ρ	0.1
$\Delta\tau_0$	2
φ	1
λ	1
η	0.9
a_1	0.5
a_2	0.5
ε	Uniform [0 1]

Table 2: Parameters used in the Firefly Algorithm.

Parameters	Value
α	0.2
β_0	0.5
γ	$\frac{1}{L}$ ($L=\max\{m,n\}$)
σ	Uniform [0,1]

by the robots. For the purposes of calculating this cost metric, some the parameters and values are summarized in Tables 1-3-2 (see [43],[46]).

In the model, e_{cc} , e_{tx} and e_{rc} have been recalculated to express them in terms of the unit of energy.

At each step of the simulation, a robot will consume an amount of energy varying its state since the robots employ different actions in different states. For example, a robot will consume more energy when handling a target than when wandering in the search area. A robot consumes 1 unit of energy for traveling from one cell to another. One stop takes an extra energy of 0.5 units. A turn of 45° takes 0.4 units of energy. Turns of 90° , 135° , 180° , take 0.6, 0.8 and 1 units of energy, respectively. These num-

Table 3: Cost related to the wireless communication.

Parameters	Value
Bit Rate (B)	3
Energy Consumed by a transceiver circuit to transmit or receive a bit, e_{cc} (Joule)	10^{-7}
Energy Consumed by a transceiver amplifier to transmit 1 bit data per meter, e_{tx} (Joule)	10^{-12}
Energy to receive a bit, e_{rc} (Joule)	10^{-7}
Path loss Exponent, ψ	[2,6]
Wireless Range R_t	6, 15

bers are approximately derived from energy measurements for Pioneer 3-DX robot [47]. We estimate the energy for performing a planned task for removing or dismantling a target is about 5 units of energy for each robot involved in the task.

The second case considers a dynamic scenario, where a complete discovery of the information from the environment could be not feasible, due to resource constraints and unpredictable events. In this case, the performance metric is given by a set of functions that measure the percentage of information related to particular objectives. For the exploration task, the metric of interested is related to the amount of unexplored map in terms of unvisited cells. While, for the case of disarmament task, the metric takes into account the percentage of detected and disarmed targets/mines. Moreover, it is considered the number of alive robots at the end of the simulations that are the robots that have not finished their budget of energy and not exploded for external events.

Regarding the simulations, there are several test-related parameters that may influence the performance and the results and they are listed as follows:

- Area size: We study both scenarios with and without obstacles. In this study, we use square areas for their symmetry and simplicity. In the future, we may experiment with nonsquare areas, or areas with no boundaries at all. (The presence of boundaries makes the problem simpler by focusing the robots on the area of interest; in a world with no boundaries, the robots could get permanently lost.)
- Robot density: This is the total number of robots in the swarm $|N^R|$.
- The number of targets $|N^T|$.
- The number of coalitions that is the minimum number of robots that can handle properly a target R_{min} .
- The transmission range R_t , which can have an effect on the recruiting task

6.2. Evaluation of the Weights under Static Conditions

In this section it is assumed that the robots have sufficient resources in terms of energy to execute the mission and the targets are static without possibility of explosion.

6.2.1. Case study 1

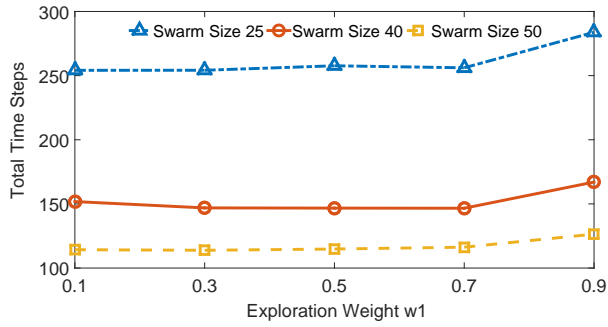
In the first set of experiments, the influence of the weights on the dimension of the area is taken into account. We have considered 50x50 square cells, 100x100 square cells, varying the team size (25, 40, 50 robots) and the number of dispersed targets. All experiments were carried out using 3 robots needed to handle a target properly.

Figures 12 and 13 show the influence of the w_1 considering different swarm size and dimension of the area evaluating respectively the total time steps to complete a mission and the total energy consumed by the swarm. It can be observed that the time steps increase as the value w_1 increases when the size of the swarm is small. This behaviour can be explained, by observing the nature of the mission that implies the collaboration of more robots in target's locations. When w_1 increases, the robots are highly motivated to explore the area. Since, the mission is completed if all target are found, motivating the robots to explore the area than disarming targets, which can lead to a temporary deadlock, especially when the swarm size is small compared to the dimension of the area and the complexity of the mission in terms of the targets. This implies the decrease of the performance of the entire system. On the other hand, a team with a larger number of robots generally increases the performance improvements. The curves do not fluctuate a lot and the total time steps is almost similar for different w_1 values. This implies that the influence of w_1 on the performance in general decreases, considering an adequate swarm size.

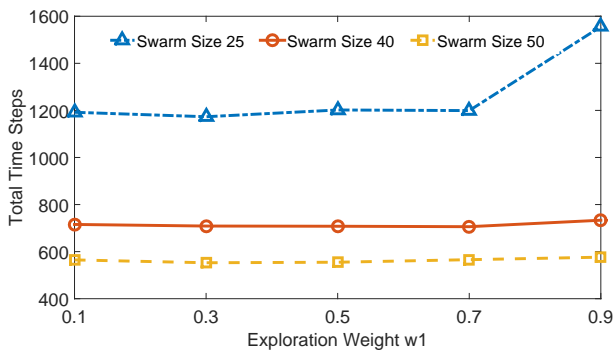
Regarding the energy consumption, it can be seen a high wastage of energy resource, considering the same total time steps, when $0.7 \leq w_1 \leq 0.9$. This difference is higher in teams with a low number of robots, compared to the number of disseminated targets $\frac{N^R}{R_{min} * N^T} \ll 1$, and in a big grid area (e.g., 30 robots operating in 100x100 cells for treating 20 targets). Considering both the total time steps and the total energy consumption, for almost all cases the best range is $0.3 \leq w_1 \leq 0.5$.

6.2.2. Case study 2

The second set of simulations compares the performance by varying the transmission range R_t (6, 15 units of cells) considering a grid area 50x50, a team with 40 robots and by varying the number of disseminated targets (15, 20, 35). This can play an important role in recruiting tasks, since for



(a)

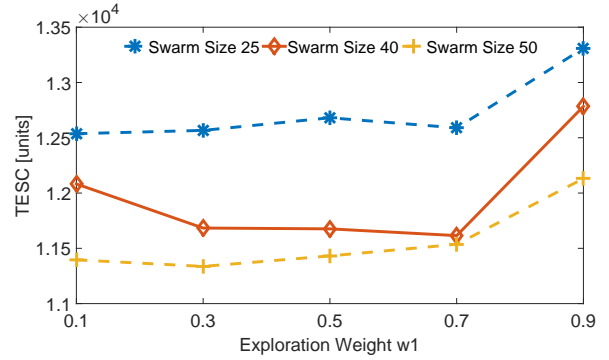


(b)

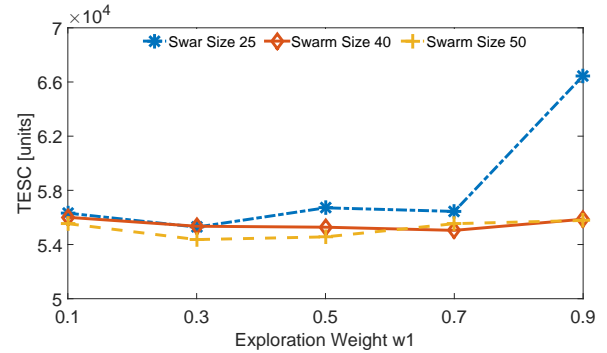
Figure 12: Evaluation of the Total Time Steps to complete the mission and 3 robots needed to handle a target. (a) 50x50 grid area, 10 targets to disarm (b) 100x100 grid area, 20 targets to be disarmed.

a higher transmission range, the probability that more robots are recruited increases. Figure 14 shows the total time steps under different conditions in terms of dispersed targets and the same swarm size (40 robots operating in the area).

It can be observed that an increase of the transmission range does not always imply the increase in performance in terms of time steps. The reason could be explained that if resources are enough in terms of robots as shown in Fig. 14 (a)-(b), an increase of the transmission range can cause some redundancy with the wastage of time to complete the mission. For example, considering a small team compared to the targets (40 robots and 35 targets), Fig. 14 (c) shows that a high transmission range with a small w_2 may imply a better performance. By increasing w_2 , more robots may be involved in the recruitment task and there is no significant difference between the two ranges. As expected, if the number of the targets to be handled is small, a high transmission range deteriorates the perfor-



(a)



(b)

Figure 13: Evaluation of the Total-Energy-System- Consumed (TESC) (a) 50x50 grid area, 10 mines to disarm and 3 robots needed to handle a target. (b) 100x100 grid area, 20 targets to disarm and 3 robots needed to handle a target.

mance since unnecessary robots could be involved in the disarmament process, depleting resources for exploration task and eventually to other targets. Although the total time steps are somewhat better, the performance in terms of energy consumed by the system strongly degrades, using a high transmission range.

Regarding the impact of w_2 on the performance, the effect of increasing the transmission range can be quantitatively notated by looking at Fig. 15. The results confirm that, especially for complex missions with $\frac{N^R}{R_{min} * NT} \ll 1$, more importance should be emphasized to the recruiting weight, thus $w_2 \geq 0.3$.

6.2.3. Case study 3

The third set of simulations investigates the effect of the weights in relation to the number of disseminated targets. The performance measures have been evaluated by varying the dimension of the area, the swarm size and the number of robots

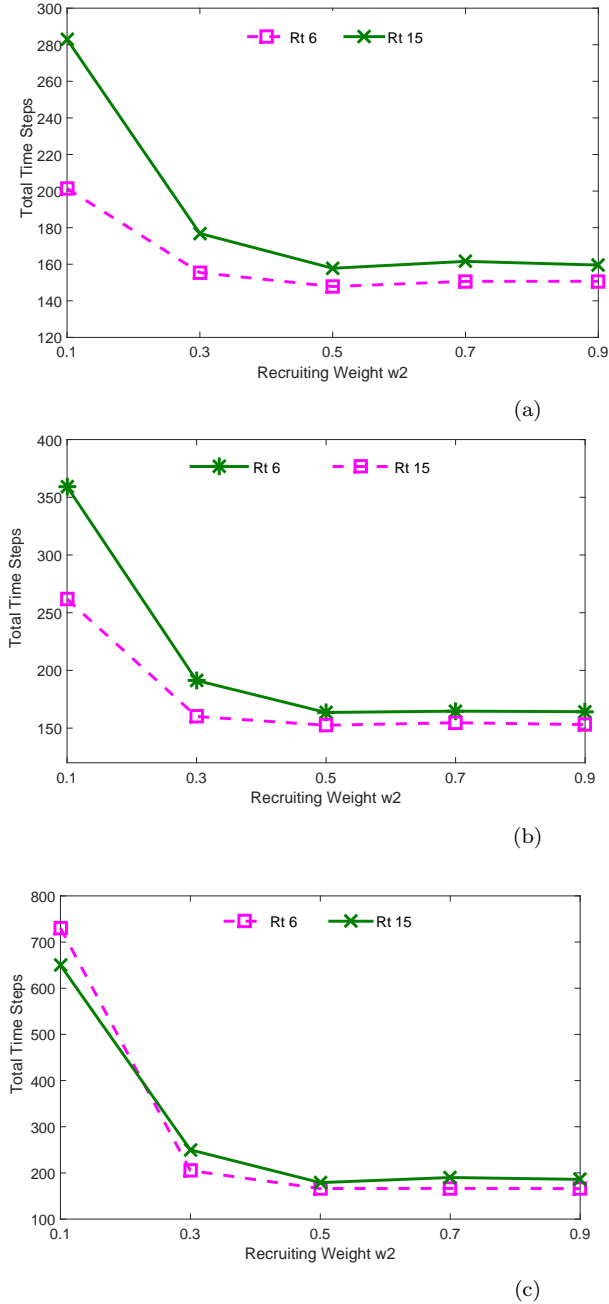


Figure 14: Evaluation of the total time steps to execute the mission in 50x50 squares and 40 robots and 3 robots to handle a target. (a) 15 targets to be handled. (b) 20 targets to be handled. (c) 35 targets to be handled.

that can deal with a target (2, 3, 4, 5).

The importance of choosing the w_2 weight properly increases as the number of the dispersed targets increases and it depends mostly on the dimension of the swarm. More specifically, if $\frac{N^R}{R_{min} * NT} \ll 1$

means the task is complex in terms of disarmament, more importance can be attributed to w_2 than w_1 as shown in Figures (16)- (17).

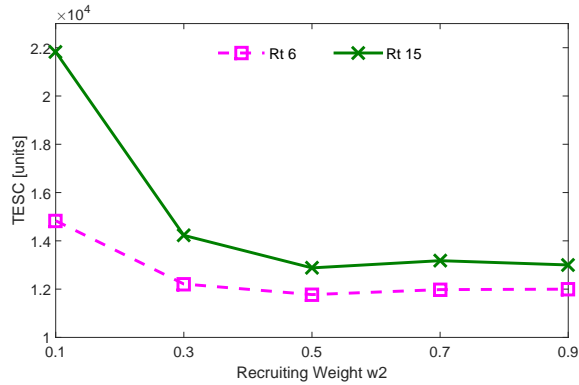
On the other hand, if $\frac{N^R}{R_{min} * NT} \approx 1$, no significant influence in terms of total time steps is observed. Obviously, more robots are introduced, less wastage of time can be observed and w_2 becomes less relevant.

In another set of experiments, we have introduced an additional parameter to control the task complexity; that is, the number of robots R_{min} required for treating a target. In this way, we can vary the task complexity and observe its influence on the impact of w_2 . Fig. 18 shows the impact of R_{min} and w_2 on the performance in terms of total time steps. It can be noticed that a high number of robots necessary to handle a target (5 robots to disarm) can cause severe resources consumption in terms of total time steps, if a small value is assigned to w_2 . This leads to weakening the ability of the robots to distribute into a target's position. As a result, the robots wondering in the area increase the time to complete the mission and the coordinators can be trapped in target's location for a long time. Thus, increasing R_{min} , the w_2 weight should be increased in order to speed up the formation of the coalition. So in this case, w_2 can greatly influence the performance and a proper value should be chosen ($w_2 \geq 0.5$). On the other hand, if the disarmament task is not particularly complex, the influence of the w_2 decreases.

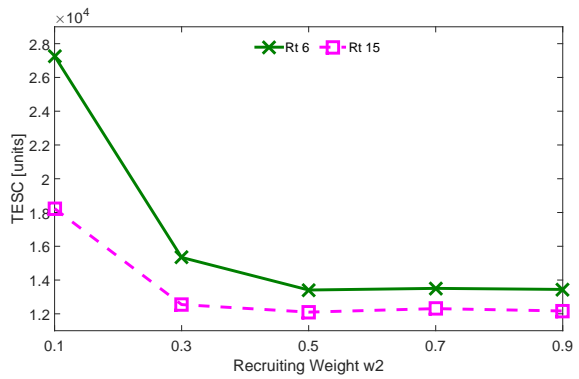
6.3. Evaluation of weights under dynamic conditions

This section investigates the effect of the weights operating in a dynamic scenario where unpredictable events can occur (such as explosion of the mines and energy constraints). It is assumed that a robot has 1000 energy units [48], without possibility of recharging during the mission, which means that if a robot consumes its energy, it will stop to perform the task at any time. In this case, to achieve good coordination and exploration is more challenging since it is required that the robot team has to respond quickly, robustly, reliably and adaptively to unexpected events.

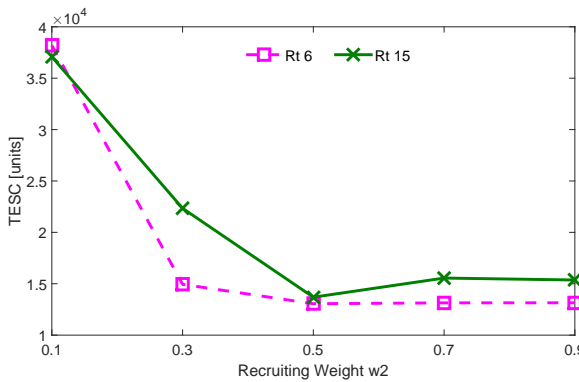
To measure the performance of such a robot team in practice, we consider a number of metrics applicable to the performance of the individual robots and the team as a whole. More specifically, we consider the percentage of unexplored cells, the num-



(a)



(b)



(c)

Figure 15: Evaluation of the total energy consumed by the system in 50x50 squares and 40 robots and 3 robots to treat a target. (a) 15 targets (b) 20 targets (c) 35 targets .

ber of disarmed targets and the percentage of alive robots.

Fig. 19 shows the impact of w_1 on the unexplored cells, varying both the dimension of the swarm and the number of dispersed targets, while keeping R_{min} as a constant. It can be noticed that for a

small robot team and a high number of targets, the performance degrades as w_1 increases. This happens because the targets, for example mines, can explode at any time, causing not only the sudden stop of some robots in nearby regions, but also the damage of possible unexplored cells that become inaccessible. In these situations, the best value is about $w_1 \leq 0.5$, which allows to balance the two tasks.

Regarding the disarmed targets, Fig. 20 highlights the impact of w_2 on the number of disarmed targets. It can be noticed that the value is particularly important for small robot teams (15 robots and 20 targets to be disarmed) and more importance can be attributed to the recruitment process ($w_2 \geq 0.3$). One possible explanation is that a robot team with higher motivation to be involved in the disarming process, can form a coalition more easily to handle a target so as to decrease the probability that it can explode.

The percentages of alive robots, evaluated considering different sizes of areas, swarm size and dispersed targets, are summarized in Fig. 21. The figure illustrates that for a small robots team, if a greater importance is given to the exploration task, some reduction of the alive robots is obtained. This behaviour seems to be influenced by the number of dispersed targets. However, increasing the swarm size leads to no significant differences. This can be justified by previous motivations; if w_1 is high, the robots may be less likely to respond to the help requests, thus leading to the coordinator robots be trapped into targets's locations waiting for others to arrive and consequently increasing the probability of some explosions. Therefore, unbalanced resources can cause severe resource wastage, lead to the potential failure of the robots due to both the energy limitation and potential explosion risks.

In almost all experiments, the performance fluctuates according to the number of disseminated targets. This may indicate that the solution, would be greatly influenced by the recruiting weight value.

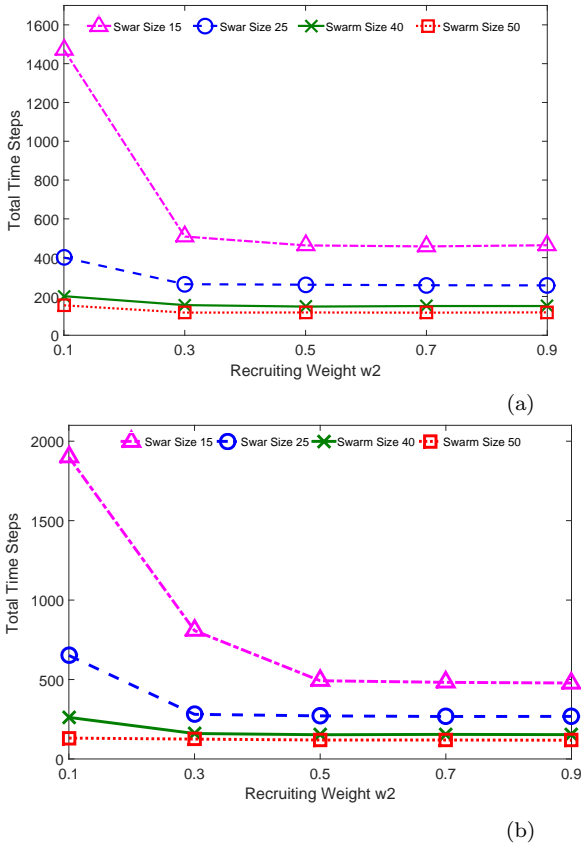


Figure 16: Evaluation of the total time steps in a grid area 50x50. (a) 15 targets. (b) 20 targets.

7. Conclusions

In this paper, we have investigated a bi-objective optimization problem for robot coordination and exploration tasks. We have considered the exploration and manipulation of hazardous targets such as mines by a swarm of robots in search and rescue mission. Specifically, we have modeled the problem as a bi-objective model and the weighted sum method is used to find a trade-off between the two tasks by varying different weighted values. The proposed strategy is bio-inspired and proves to be robust, effective with the use of limited resources in a balanced way.

In order to test the validity of our proposed model and the influence of the weight values, a Java-based simulator has been developed and implemented. Different experimental scenarios have been considered to suitably evaluate the impact of the weight values on the critical parameters of the problem such as the dimension of the area, num-

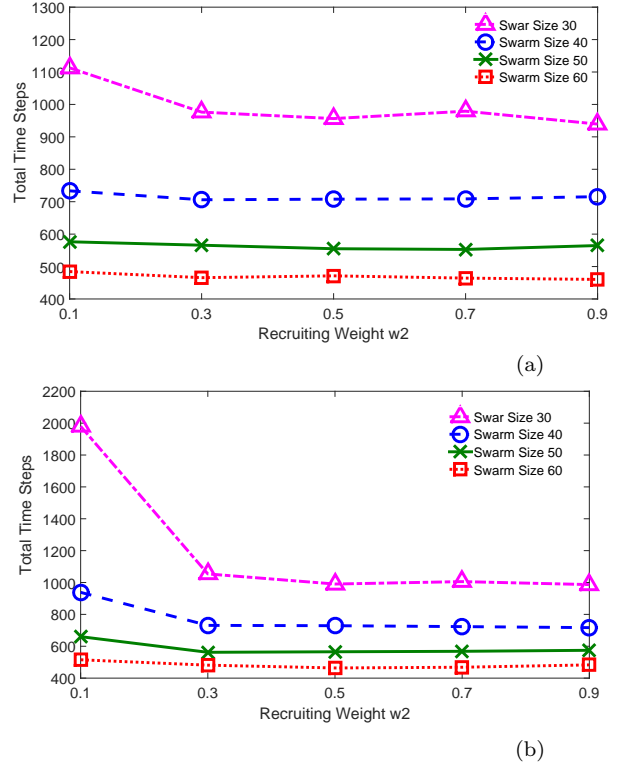


Figure 17: Evaluation of the total time steps in a 100x100 grid area, varying the dimension of the swarm with (a) 20 targets and (b) 30 targets.

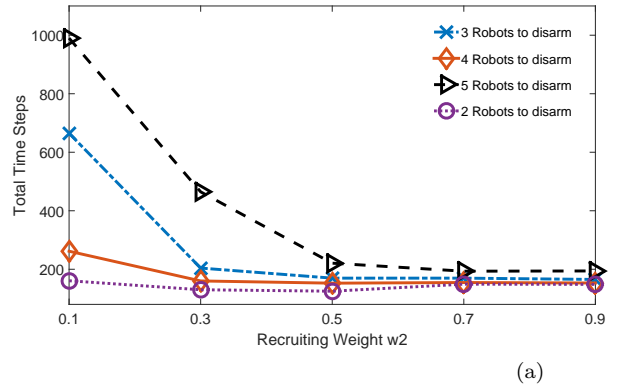
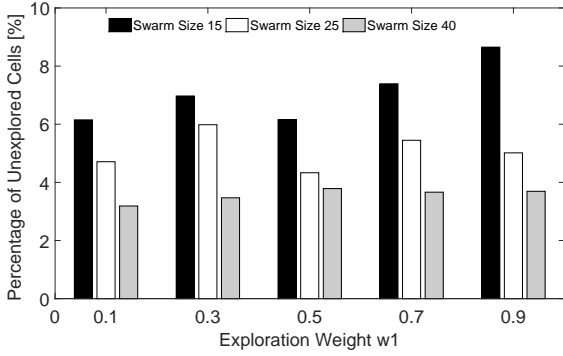
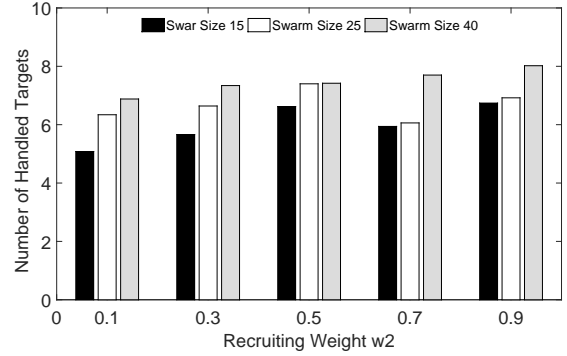


Figure 18: Evaluation of the total time steps varying R_{min} in an area with 50x50 squares, 40 robots and 20 targets.

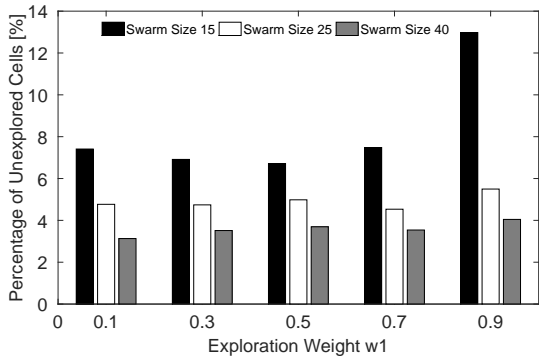
ber of disseminated targets and number of robots to coordinate. The results have demonstrated that the choice of the right compromise between the two tasks is not straightforward. Generally speaking, the proper values depend on the application context. In most cases, the trade-off between the two objectives is highly correlated with the number of



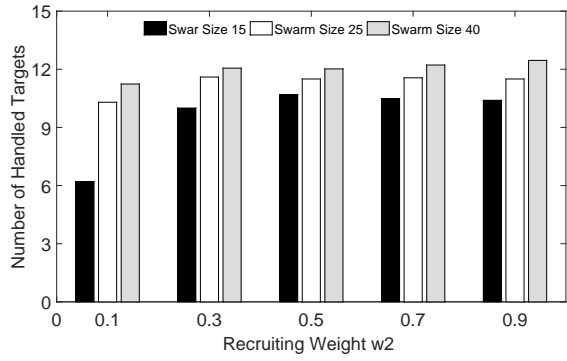
(a)



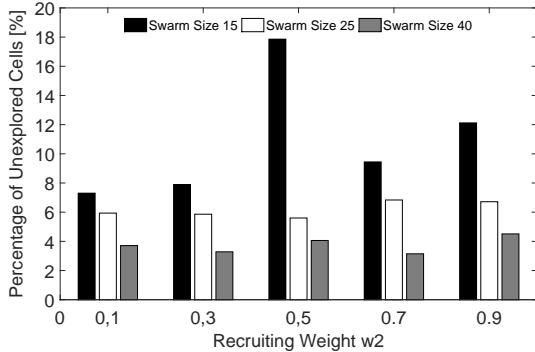
(a)



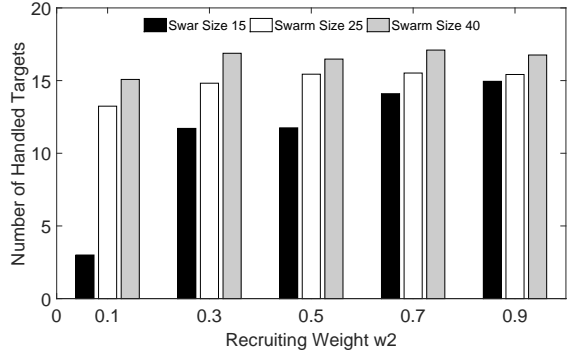
(b)



(b)



(c)



(c)

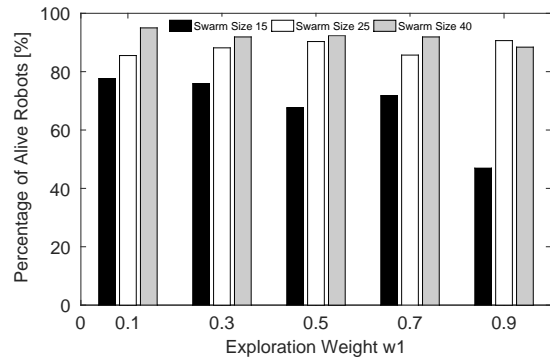
Figure 19: Percentage of unexplored cells in a 50x50 grid area, varying the dimension of the swarm of robots and 3 robots needed to handle a target. (a) 10 targets (b) 15 targets (c) 20 targets

Figure 20: Percentage of disarmed targets in a 50x50 grid area, varying the dimension of the swarm of robots and 3 robots needed to handle a target. (a) 10 targets (b) 15 targets (c) 20 targets

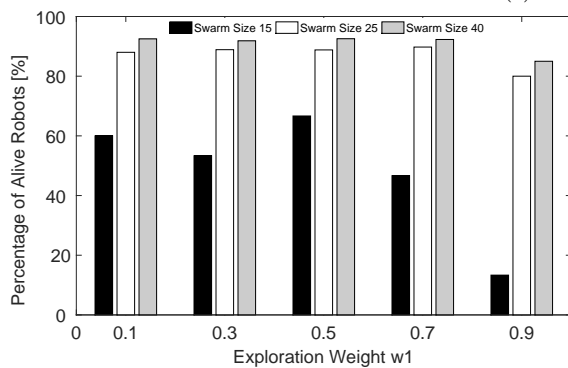
targets dispersed in the area, compared to the dimension of robot swarm. In these cases, more importance should be attributed to w_2 so that ($w_2 \geq 0.5$). However, in general case, balanced weights w_1 and w_2 (around 0.5) can offer a better trade-off.

Possible future works include the extension of methods to dynamically adjust the weights during the mission so as to be adaptive to the resource

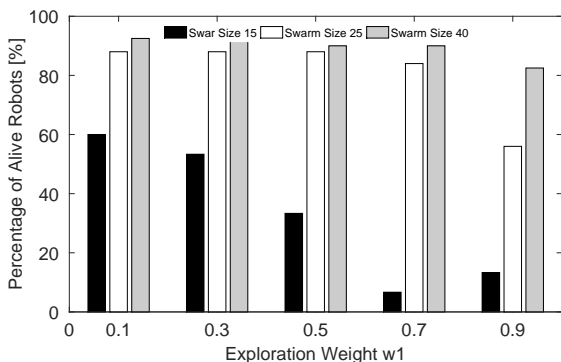
of the robots or other constraints. Furthermore, it will be useful to consider more realistic models and practical issues that robots may face in real-world scenario such as sensing and communication failure. In addition, the proposed method can be modified to potentially deal with the unknown but mobile targets in an unknown area, which will be explored further in future work.



(a)



(b)



(c)

Figure 21: Percentage of alive robots in a 50x50 grid area, varying the dimension of the swarm of robots and 3 robots needed to handle a target. (a) 10 targets (b) 15 targets (c) 20 targets

8. Conflict of Interest

The authors declare that they have no conflict of interest.

References

- [1] L. Parker, Distributed intelligence: Overview of the field and its application in multi-robot

system, *Physicl Agents* 1 (11) (2008) 5–14. doi:10.1016/0003-4916(63)90068-X.

- [2] A. Khamis, A. Hussen, A. Elmology, Multi-robot task allocation: A review of the state-of-the-art, in: *Cooperative Robots and Sensor Networks*, Springer International Publishing, 2015, pp. 31–51. doi:10.1007/978-3-319-18299-5_2.
- [3] X.-S. Yang, *Nature-inspired Optimization Algorithms*, Elsevier Insight, 2014.
- [4] M. Dorigo, T. Stutzle, The ant colony optimization metaheuristic:algorithm, applications, and advance, in: *Handbook of Metaheuristic*, Springer US, 2003, pp. 250–285.
- [5] X.-S. Yang, Firefly algorithms for multimodal optimization, in: *Stochastic algorithms:foundations and applications*. SAGA, Vol. 5792, Springer, 2009, pp. 169–178.
- [6] N. Hoff, A. Sagoff, R. Wood, R. Nagpal, Two foraging algorithms for robot swarm using only local communication, in: *IEEE International Conference on Robotics and Biomimetic*, 2010.
- [7] Y. Zhang, D. Gong, J. Zhang, Robot path planning in uncertain environment using multi objective particle swarm optimization, *Neurocomputing* 103 (2013) 172–185.
- [8] M. Dadgar, S. Jafari, A. Hamzeh, A pso based multi robot cooperation method for target searching in unknown environment, *Neurocomputing* 177 (2016) 62–74.
- [9] Y. Quionez, J. de Lope, D. Maravall, Cooperative and competitive behaviors in a multi robot systems for surveillance tasks, in: *International Conference on Computer Aided Systems Theory EUROCAST*, 2009, pp. 437–444.
- [10] S. Takadoro, *Rescue Robotics DDT Project on Robots and Systems for Urban Search and Rescue*, Springer-Verlag London, 2009.
- [11] N. Basilico, F. Amigoni, Exploration strategies based on multi-criteria decision making for searching environments in rescue operations, *Auton Robot* (3) (2011) 401–417.
- [12] N. Geng, D. W. Gong, Y. Zhang, Pso-based robot path planning for multisurvivor rescue in limited survival time, *Mathematical Problems in Engineering* 2014.
- [13] F. D. Rango, N. Palmieri, Ant-based distributed protocol for coordination of a swarm of robots in demining mission, in: *Proc. SPIE, Unmanned Systems Technology*, 2016.
- [14] A.Khamis, A.ElGindy, Minefield mapping using cooperative multirobot systems, *Journal of Robotics* 2012.
- [15] L. Parker, Decision making as optimization in multi-robot teams, in: *International Conference on Distributed Computing and Internet Technology ICDCIT*, Springer, Berlin, Heidelberg, 2012.
- [16] M. A. Hsieh, A. Halsz, S. Berman, V. Kumar, Biologically inspired redistribution of a swarm of robots among multiple sites, *Swarm Intelligence* 2 (2008) 121–141.
- [17] T. Kuyucu, I. Tanev, K. Shimohara, Evolutionary optimization of pheromone-based stigmergic communication, in: *European Conference on the Applications of Evolutionary Computation*, Springer, Berlin, Heidelberg, 2012.
- [18] R. Calvo, J. R. de Oliveira, M. F. R. Romero, Bio-inspired coordination of multiple robots systems and stigmergy mechanisms to cooperative exploration and surveillance tasks, in: *IEEE 5th International Con-*

- ference on Cybernetics and Intelligent Systems (CIS), 2011.
- [19] S. Doi, Proposal and evaluation of a pheromone-based algorithm for the patrolling problem in dynamic environments, in: *Swarm Intelligence (SIS)*, 2013 IEEE Symposium on, 2013, pp. 48–55.
- [20] J. L. Pearce, B. Powers, C. Hess, P. E. Rybski, S. Stoeter, N.Papanikolopoulos, Using virtual pheromones and cameras for dispersing a team of multiple miniature robots, *Journal of Intelligent and Robotic Systems* 45 (2006) 307–321.
- [21] J. Osherovich, V. Yanovski, I. A. Wagner, A. M. Bruckstein, Robust and efficient covering of unknown continuous domains with simple, ant-like a(ge)nts, *The International Journal of Robotics Research* 27 (2008) 815831.
- [22] B. Ranjbar-Sahraei, G.Weiss, A. Nakisaee, A multi-robot coverage approach based on stigmergic communication, in: *German Conference on Multiagent System Technologies MATES*, 2012, pp. 126–138.
- [23] A. Ravankar, A. Ravankar, Y. Kobayashi, T. Emaru, On a bio-inspired hybrid pheromone signalling for efficient map exploration of multiple mobile service robots, *Artificial Life and Robotics* 21 (2016) 221231.
- [24] J. de Lopea, D. Maravalla, Y.Quinonezb, Self-organizing techniques to improve the decentralized multi-task distribution in multi-robot systems, *Neurocomputing* 163 (2015) 4755.
- [25] Y. Yang, C. Zhou, Y. Tian, Swarm robots task allocation based on response threshold model, in: *Autonomous Robots and Agents, ICARA 4th International Conference on*, 2009.
- [26] S. Triguia, et al., A distributed market-based algorithm for the multi-robot assignment problem, *Procedia Computer Science* 32 (2014) 1108–1114.
- [27] E. G. Jones, M. B. Dias, A. Stentz, Learning-enhanced market-based task allocation for oversubscribed domains, in: *Intelligent Robots and Systems, IROS, IEEE/RSJ International Conference on*, 2007. doi:10.1109/IROS.2007.4399534.
- [28] K. Derr, M. Manic, Multi-robot, multi-target particle swarm optimization search in noisy wireless environments, in: *IEEE, Human System Interactions, HSI '09. 2nd Conference on*, 2009.
- [29] J. L. et al., A modified particle swarm optimization algorithm for distributed search and collective cleanup, in: *IEEE, Awareness Science and Technology and Ubi-Media Computing (iCAST-UMEDIA), International Joint Conference on*, 2014.
- [30] P. Das, et al., A hybrid improved pso-dv algorithm for multi-robot path planning in a clutter environment, *Neurocomputing* 207 (2016) 735–753.
- [31] M. S. Couceiroa, P. A. Vargasc, R. Rochaa, N. Ferreirab, Benchmark of swarm robotics distributed techniques in a search task, *Robotics and Autonomous Systems* 62 (2014) 200–213.
- [32] S. Momen, Ant-inspired decentralized task allocation strategy in groups of mobile agents, *Procedia Computer Science* 20 (2013) 169–176.
- [33] J. Banharnsakun, T. Achalakul, R. C. Batra, Target finding and obstacle avoidance algorithm for microrobot swarms, in: *Systems, Man, and Cybernetics (SMC), IEEE International Conference on*, 2012.
- [34] J. Liang, C. Lee, Efficient collision-free path-planning of multiple mobile robots system using efficient artificial bee colony algorithm, *Advances in Engineering Software* 79 (2015) 47–56.
- [35] M. P. Garciaa, et. al., Path planning for autonomous mobile robot navigation with ant colony optimization and fuzzy cost function evaluation, *Applied Soft Computing* 9 (2009) 1102–1110.
- [36] F. Arvin, et. al., Cue-based aggregation with a mobile robot swarm: a novel fuzzy-based method, *Adaptive Behavior* 22 (2014) 189–206.
- [37] N. Palmieri, F. de Rango, X. S. Yang, S. Marano, Multi-robot cooperative tasks using combined nature-inspired techniques, in: *Computational Intelligence (IJCCI), 7th International Joint Conference on*, 2015.
- [38] L. Bayindir, A review of swarm robotics tasks, *Neurocomputing* 172 (2016) 292–321.
- [39] M. Senanayakea, et al., Search and tracking algorithms for a swarm of robo: a survey, *Robotics and Autonomous Systems* 75 (2016) 422–434.
- [40] V. Barichard, X. Gandibleux, V. T'Kindt, *Multiobjective Programming and Goal Programming Theoretical Results and Practical Applications*, Springer-Verlag Berlin Heidelberg, 2009.
- [41] R. Marler, J. Arora, The weighted sum method for multi-objective optimization: new insights, *Structural and Multidisciplinary Optimization* 41 (2010) 853–862.
- [42] Y. Collette, P.Siarry, *Multiobjective Optimization Principles and Case Studies*, Springer-Verlag Berlin Heidelberg, 2004.
- [43] C. C. Ooi, C. Schindelbauer, Minimal energy path planning for wireless robots, *Mobile Networks and Applications* 14 (2009) 309–321.
- [44] C. Tang, P. K. McKinley, Energy optimization under informed mobility, *IEEE Transaction on parallel and distributed Systems* 17 (2009) . 947–962.
- [45] N. Palmieri, X.-S. Yang, F. De Rango, S. Marano, Comparison of bio-inspired algorithms applied to the coordination of mobile robots considering the energy consumption, *Neural Computing and Applications*doi:10.1007/s00521-017-2998-4.
- [46] N. Palmieri, S. Marano, Discrete firefly algorithm for recruiting task in a swarm of robots, in: *Nature-Inspired Computation in Engineering*, Springer International Publishing, 2015, pp. 133–150. doi:10.1007/978-3-319-30235-5_7.
- [47] Y. Mei, Y. Lu, C. Lee, Y. Hu, Energy efficient mobile robot exploration, in: *Robotics and Automation, ICRA. Proceedings IEEE International Conference on*, 2006.
- [48] T. W. et al., Staying-alive and energy-efficient path planning for mobile robots, in: *American Control Conference, IEEE*, 2008.