2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

Multi-Robot Containment and Disablement

Yuval Maymon[1] and Noa Agmon[2]

***Abstract*— This paper presents the multi-robot containment and disablement (CAD) problem. In this problem, a team of (ground or aerial) robots are engaged in a cooperative task of swarm containment and disablement (for example, locust swarm). Each team member is equipped with a tool that can both detect and disable the swarm individuals. The swarm is active in a given physical location, and the goal of the robots is twofold: to contain the swarm members such that the individuals will be prevented from expanding further beyond this area (this is referred to as perfect enclosure), and to fully disable the locust by reducing the size of the contained area (while preserving the perfect enclosure). We determine the minimal number of robots necessary to ensure perfect enclosure, and a placement of the robots about the contained area such that they will be able to guarantee perfect enclosure, as well as a distributed area reduction protocol maintaining perfect enclosure. We then suggest algorithms for handling the case in which there are not enough robots to guarantee perfect enclosure, and describe their performance based on rigorous experiments in the TeamBots simulator.**

I. INTRODUCTION

Robot teams are considered for use in various tasks such as search and rescue, security, and delivery. Recently, robots are becoming more prevalent also in agriculture [1] for missions such as dealing with resistance weed [2], field work [3], and pest control [4]. In our work we are motivated by the problem of disabling locust swarm or other mobile pests, that cause severe damage to crops and are considered a devastating natural disaster.

We therefore define a new robotic problem, multi-robot containment and disablement, in which a team of robots are engaged in a cooperative task of containing and disabling a swarm of mobile entities, in our case locust swarm. Each team member is equipped with a tool that can both detect and disable the swarm individuals. The locust swarm is mobile and active in a given physical area, and the goal of the robots is twofold: to contain the locust such that the swarm will be prevented from expanding further beyond this area (causing additional damage), and minimize the time to disabling all swarm members. The problem draws similarities to several canonical robotic problems: robotic coverage, enclosure, and convergence, yet it raises complex, innovative, challenges that are handled herein.

The solution concept proposed in this paper is composed of two stages: spread the robots around the area guaranteeing that no locust swarm member will be able to leave the area undetected (we refer to this as *Perfect Enclosure*), and incrementally move the robot team members towards one another, eventually meeting, and by that covering the contained area and disabling all locust swarm members. We show that these two tasks are twine together, and specifically finding the minimal number of robots to enclose an area might not result in the ability to provide a finite upper bound on the meeting time.

We determine the number of robots and their placement around the borders of this area guaranteeing perfect enclosure *and* meeting, and show correlation to an upper bound on the meeting time of the robots, as well as a distributed protocol guaranteeing that the robots meet and maintain perfect enclosure while they do so. We then suggest different protocols for handling the case in which there are not enough robots to guarantee perfect enclosure, referred to as *Imperfect Enclosure*, and examine those empirically using the TeamBot simulator.

*This work was not supported by any organization

[1]Yuval Maymon is with the Computer Science Department, Bar-Ilan University, Ramat Gan 5290002, Israel yuvalmaymon30@gmail.com

[2]Noa Agmon is with the Computer Science Department, Bar-Ilan University, Ramat Gan 5290002, Israel agmon@cs.biu.ac.il

II. Related Work

The CAD problem is strongly related to different canonical problems in multi-robot systems, such as coverage (mainly refers to an area/path coverage) [5-8], enclosure (a robots formation around a specific area) [9-11] and gathering (robots gathering together to perform a task) [12]. In this section we briefly stroll through these and other research areas, and explain the unique challenges of the CAD problem compared to them.

Lien et al. [13] examine the sheep herding problem, in which a group of robots cooperate in order to control the motion of a flock, using no communication. They suggest several behaviors, based on team formation, for successfully steering the flock to a desired location. Pierson and Schwager [14] also consider the herding problem by multiple robots for relocating the herd to a desired location, assuming a non-cooperative herd with non-linear response of the individuals to the herding robots. They focus on a control strategy for the robots using repulsive and attractive forces. Cowling and Gmeinwieser [15] examine the herding problem with one robot travelling in circling paths for guiding the sheep to a certain location. Evered et al. [16] examine empirically practical aspects of using a robot to herd sheep, concluding that the sheep quickly adapt to robot presence. All these concentrate on leading the flock to a desired location, while we focus on containment *and* disablement.

Cheng et al. [17] examine the containment problem, presenting how leader robots contain followers robots and proceed to a predefined target. Guo et al. study the *moving-target-enclosing* problem [10], and show that with only local measured information, the robots asymptotically form a formation enclosing the target. Mehendale [18] presents a containment of randomly located singular adversarial agents using potential functions.These approaches do not examine the minimal number of robots achieving containment, nor do they strive to meet.

Kubo et al. [19] analyze conditions in which a robotic swarm with a simplest communication device can succeed to enclose a target. In our research we assume perfect communication between a robot and its closest neighbours, and we focus on containment of multiple agents and not just one.

A relatively new problem concerning the ability to externally effect the behavior of a swarm comes from the field of *ad-hoc teamwork* [20], [21]. In this problem, a team of more informed agents (or robots) attempt to lead the swarm members into acting in a certain way that will result in a better group utility.In our work, however, the robots do not share the same goals of the swarm agent but they have conflicting goals.

The CAD problem resembles also the coverage problem, in which a team of robots are required to jointly visit all points in a given area [5-8]. Karl Obermeyer et al. [8] propose a distributed coverage algorithm for the *Art Gallery Theorem with Holes* problem. He assumes the robots have a line-of-sight sensing and provides an upper bound of $n+2h-2$ on the number of robots, where n is the number of vertices of the polygon and h is the number of holes. The CAD problem is slightly different because the robots have a field-of-view sensing, rather than line-of-sight sensing. Also in the CAD problem the robots should cover together all points in the area for disabling all swarm members. However, since the swarm agents are mobile, simply covering the area will not suffice: the swarm agents might spread further and incrementally increase the area to be covered. Thus we must maintain perfect enclosure as well as covering the area.

Li Huang et al. [22] survey multi-robot adversarial patrolling problems. As in [22], we try patrolling in a cyclic path around a closed polygon. Moreover, we propose a new approach of minimizing this polygon in a cyclic path too.

III. CAD Problem Definition

We are given a team of n homogeneous robots $R = \{r_1, \ldots, r_n\}$ engaged in a cooperative task of locust-swarm enclosure and disablement. The robots may be grounded, or aerial. Each robot is equipped with a tool that allows it to detect and disable swarm agents within a range of $360°$, where for aerial robots the detection and disablement is in a cone projection of angle α, thus we refer to its associated 2D circle projection of radius d. We denote this 2D circle projection of robot r_i as its *field of view*, or $\texttt{fov}_i$ (wherever possible, we will simply use the term fov). For ground robots, this refers to the 2D sensing/disablement circle of radius d. We assume that the communication range of each robot is limited, but it can at least communicate with robots in distance of $2d$ from it. We also assume that the robots maintain collision avoidance between themselves.

Given a group of m locust agents $A = \{a_1, \ldots a_m\}$ (agents, in short), each a_i located at time t in point p_i^t, where $P_t = \{p_1^t, \ldots, p_m^t\}$. First, we would like to determine the minimal number of robots and their location (a polygon) guaranteeing that all locust agents in the environment will be contained in time 0 within the fov of all robots, without the ability to escape. That is, an agent will not be able to leave the area without being detected/disabled by some robot. This is referred to as *Perfect Enclosure*. Next, given that the robots are placed around the determined polygon P, they should reduce its area while guaranteeing the perfect enclosure is maintained and eventually the robots will meet, that is, $P \setminus \{\bigcup_{j=1}^{n} \bigcup_t \texttt{fov}_j^t\} = \emptyset$, where $\texttt{fov}_j^t$ is the fov of robot j in time t. Therefore, a solution to CAD should fulfill the following two objectives:

1) Guarantee that the initial enclosure is perfect, and will remain perfect even if the robots are moving.
2) The time to fully reduce the area, denoted at *meeting time*, is minimized.

The CAD problem is examined in Section IV. If the number of robots is not enough to guarantee perfect enclosure, this is referred to as *Imperfect Enclosure*, and the goal of the robots would be to maximize the percentage of disabled locust agents. We present an empirical analysis of this case in Section V.

IV. Perfect Enclosure

In this section we first examine the basic CAD problem, that is, finding the minimal number of robots guaranteeing perfect enclosure, and determining the initial placement of the robots for that purpose. We start by determining the structure of the area for containment that will require the minimal number of robots for perfect enclosure (with respect to P_0).

We would like to find the structure minimizing the number of robots necessary to cover the border of the structure (enclose it). Considering continuous shapes, we compare between two options: the minimal enclosing polygon of the location of the agents (denoted as $\texttt{MEP}(P_0)$) or the convex hull of those points (denoted as $\texttt{CH}(P_0)$). Denote by $rmin(P)$ the minimal number of robots with fov of radius d necessary to fully cover a path P, that is, each point in P is in the fov of some robot from the team. Note that we examine placements of robots only on the circumference of the polygon, since finding the minimal number of robots to fully cover a polygon, or even a boundary area of a polygon, is a hard problem [23].

The CH requires less robots than the MEP on the concave segments (due to triangle inequality). However, if there are overlaps between fovs of robots on different edges (such as a sharp star shape), $rmin(\texttt{MEP})$ may be smaller than $rmin(\texttt{CH})$. As seen in our experiments, these cases are rare, thus in this paper we focus our analysis on CH.

Based on this initial placement, the robots move towards the center of the polygon while maintaining the perfect enclosure and allowing the robots to progress in a way that will eventually result in a complete disablement of the locust agents.

A. Initial Robot Placement

Given the convex polygon which contains the area, the robots are placed along the edges of the polygon. We assume that the robots start on a random places and converge to their initial location on the containing polygon. We also assume that the containing polygon is shared with all robots before they start moving. The robots know when they should start moving by a simple synchronization protocol shared with their neighbours.

In order to obtain perfect enclosure, it is sufficient to place the robots along the polygon such that the distance between two adjacent robots is $\leq 2d$ (clearly, a distance $> 2d$ will violate the perfect enclosure). Denote the length of the circumference of a polygon P by $len(P)$. The minimal number of robots required for a perfect enclosure of the initial polygon, $rmin(P_0)$ equals $\lceil \frac{len(P_0)}{2d} \rceil$. However, we show in Lemma 1 that if the intersection between the fovs of two neighboring robots along the polygon is one point, then it is impossible to guarantee both that the perfect enclosure is maintained, and provide a finite bound on the meeting time. This means that even if we find the minimal number of robots to guarantee perfect enclosure, it might not be enough to solve the CAD problem. Denote the distance between two points p and q by $dist(p,q)$.

Lemma 1. *Given a team of n robots, each with fov of radius d, placed along a convex polygon P with such that the fov of every two adjacent robots r_i and r_j along the polygon touch at a point $l_{(i,j)}$. There is no distributed algorithm $\mathcal{A}$ that can guarantee convergence within time $t_B < \infty$ while maintaining perfect enclosure.*

Proof. Let r_a, r_b and r_c be three adjacent robots along the circumference of the polygon, such that the fovs of r_a and r_b touch at $l_{(a,b)}$, and the fovs of r_b and r_c touch at $l_{(b,c)}$. We first consider the case in which r_a, r_b and r_c are on the same edge. Without loss of generality, let r_b move a distance of ϵ to the point z either perpendicular to the polygon edge, or in an angle $\angle r_c r_b z = \beta < \pi$. Thus, from the law of cosines on the triangle $\triangle r_c z r_b$ it follows that $dist(r_b, r_a) \geq \sqrt{(2d)^2 + \epsilon^2} > 2d$, hence the fovs of r_a and r_b are detached, breaking the perfect enclosure. Therefore r_b has to move in coordination with r_a and r_c. However, similar argument holds for both robots adjacent to r_a and r_c, and to their adjacent neighbors, and so on. Therefore the only algorithm guaranteeing maintaining perfect enclosure is one that schedules all robots to travel in unison, which is impossible for a distributed control algorithm for robots with communication range of $2d$.

Therefore assume that r_a, r_b and r_c are not on the same edge. We would like to show that even in the case in which r_b proceeds more than the overlap area, then still the distance it passes is $\epsilon \to 0$, hence the robots can not proceed and the algorithm will not converge. Without loss of generality, let r_a be on edge e_i and r_b and r_c be on other edge e_j. The case in which $l_{(a,b)}$ is not the intersection between e_j and r_b is not possible since the perfect enclosure will not be maintained.

Since r_a and r_b are not on the same edge, and since the perfect enclosure must be maintained, $l_{(a,b)}$ must be the intersection point between e_j and r_b (see Fig. 1). Denote the angle between e_i and e_j as α, and the intersection point as C.

Consider first the case in which $\alpha = \pi - \epsilon$, where $\epsilon \to 0$, thus $\alpha \to \pi$.

Robot r_b can move in a direction between 0 to $\pi/2$. Robot r_b can not move outside of the overlap area between its fov and r_a's fov, otherwise the perfect enclosure will not be maintained. Denote as $r_{b'}$ the center of r_b after it proceeded in a perpendicular direction to edge e_j, until its fov is touching r_a's fov. Note that $|\overline{r_b r_{b'}}|$ is the longest distance r_b can proceed until it is detached from r_a.

Since $\alpha \to \pi$ then $\sphericalangle l_{(a,b)} r_a C = \beta \to 0$. By angle rules, we get that $\sphericalangle r_a l_{(a,b)} r_b = \alpha + \beta$. $l_{(a,b)}$ is on the fov's of r_a and r_b, hence $|\overline{r_a, l_{(a,b)}}| = |\overline{r_b, l_{(a,b)}}| = d$. By the cosine rule of the non-right triangle $r_a l_{(a,b)} r_b$ we get:

$|\overline{r_a, r_b}|^2 = \lim_{\alpha \to \pi \beta \to 0} d^2 + d^2 - 2 * d * d * \cos(\alpha + \beta) = \lim_{\alpha \to \pi \beta \to 0} 2d^2 - 2d^2 * \cos(\alpha + \beta) = 4d^2$.

We get $|\overline{r_a, r_b}| = 2d$.

Since the triangle $r_a l_{(a,b)} r_b$ is an isosceles triangle, $\sphericalangle l_{(a,b)} = (\pi - \alpha - \beta)/2$. Also, $\sphericalangle C r_b r_{b'} = \pi/2$, we get $\sphericalangle r_{b'} r_b r_a = (\alpha + \beta)/2$. The triangle $r_{b'} r_b r_a$ is also an isosceles triangle because $|\overline{r_a, r_{b'}}| = 2d$, so $\sphericalangle r_{b'} r_a r_b = \pi - \alpha - \beta$. By the cosine rule we get:

$|\overline{r_{b'} r_b}|^2 = \lim_{\alpha \to \pi \beta \to 0} 4d^2 + 4d^2 - 2 * 4d^2 * \cos(\pi - \alpha - \beta) = 8d^2 - 8d^2 * \cos(0) = 0$.

We get that the size of $r_b r_{b'} = \epsilon \to 0$ as $\alpha \to \pi$. □

Therefore, the robots are placed with a fov-overlap of at least $0 \leq x \leq d$ (see illustration in Fig. 2). Hence the minimal number of robots obtaining the fov-overlap of x is $rmin_x(P_0) = \lceil \frac{len(P_0)}{2d - x} \rceil$. Note that if $x \geq d$ then the contribution of at least one robot is redundant, and thus can be removed. The value of x determines the speed of

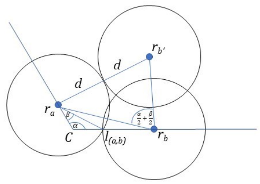

Fig. 1: Illustration of Lemma 1

convergence.

B. Area Reduction

As established in the previous section, our solution concept is based on the fact that we placed the robots around the agents' initial locations. Note that if we place all robots in a way that will create a complete coverage of the boundaries of initial polygon, then any agent attempting to move outside of the polygon will be disabled.

After creating the perfect enclosure, all robots progress into the polygon such that they maintain perfect enclosure of the area. Therefore, no robot can travel too far into the polygon to avoid breaking the enclosure. The direction and distance travelled are defined as follows:

Direction The movement direction is defined to be the perpendicular line to the edge which the robot is located on.

Distance The robot moves as far as it can while maintaining a `fov`-overlap with its neighbours (any overlap). We denote this distance by $\rho(x)$. That is, $\rho(x)$ = The distance between two intersection points of two adjacent robot's `fovs` (see illustration in Fig. 2).

Let r_1 and r_2 be two adjacent robots along the enclosing polygon, and denote the two intersection points between their `fovs` by A and B. We define the maximal distance traveled by r_1 while still maintaining `fov`-intersection by $\rho(x)$, that is, $\rho(x) = |\overline{AB}|$.

Lemma 2. $\rho(x) = 2\sqrt{dx - \frac{x^2}{4}}$

Proof. Let C be the middle point on the line AB, that is, $dist(A, C) = dist(C, B)$. Therefore $|AB|= 2|AC|$. Since all `fovs` have the same radius d, C is the middle point of the line between the centers of the `fovs` H_1 and H_2. $|H_1H_2|= 2d - x$, therefore we get that $|H_1C|= d - \frac{x}{2}$. The angle between the H_1H_2 and AB is $\frac{\pi}{2}$, thus by Pythagorean Theorem we get: $|AC|^2+|H_1C|^2= |AH|^2$

$\Rightarrow |AC|^2+(d - \frac{x}{2})^2 = d^2$

$\Rightarrow |AC|= \sqrt{dx - \frac{x^2}{4}}$

Finally we get: $\rho(x) = |AB|= 2|AC|= 2\sqrt{dx - \frac{x^2}{4}}$ □

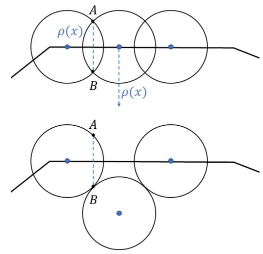

Fig. 2: Illustrating the direction and distance ($\rho(x)$) travelled by a robot at each time step in a fraction of the enclosing polygon.

Algorithm `ReduceArea` describes the area reduction procedure, leading the full reduction of the area. Note that each robot needs to communicate only with its two adjacent robots during the execution of the algorithm. Therefore, this can be executed in a completely distributed manner, once the robots arrive at their initial positions. Once the robots move towards the center of the polygon, the size of the polygon decreases, and thus some robots are no longer used for the area reduction procedure. Formally, a robot r_i is *redundant* if the `fovs` of its both adjacent neighbors along the polygon r_{i-1}, r_{i+1} are overlapping, or it is outside the containing polygon. The time complexity of the algorithm is $\mathcal{O}(1)$.

Theorem 1. *Given a team of n robots with `fov` $=$ d and traveling in velocity v placed around a convex polygon P of diameter D_p, with `fov`-overlap x, $0 < x \leq d$, and network connection time w. Then if the robots follow Algorithm `ReduceArea`, it is guaranteed that they maintain perfect enclosure at all time step $t > 0$, and that the upper bound on the*

Algorithm 1 `ReduceArea`(i, i_e, x, n)

i = index of the robot by the placing order
i_e = index of the robot on the edge
x = `fov`-overlap distance
n = number of robots

$next$ = robot number $(i + 1)\%n$
$previous$ = robot number $(i - 1)\%n$
if robot i is not redundant **then**
  if i_e is even **then**
    if $next$ and $previous$ are standing **then**
      Move forward $\rho(x)$ distance
      Stand
    end if
  else
    if $previous$ has moved and standing **then**
      Move forward $\rho(x)$ distance
      Stand
    end if
  end if
end if

convergence of the polygon P is $\frac{D_p}{v} + \frac{D_p * w}{2\sqrt{dx - \frac{x^2}{4}}}$ *time units.*

Proof. Let $round$ be the time it takes one robot to proceed a distance of $\rho(x)$ and sending a message to its neighbours to start moving. By the $time * velocity = distance$ equation we get that a robot passes a distance $\rho(x)$ in $\frac{\rho(x)}{v}$ time units. We get that $round = \frac{\rho(x)}{v} + w$. Following Lemma 2, at the first $round$ a robot can move towards the center of the polygon a distance of $\rho(x) = 2\sqrt{dx - \frac{x^2}{4}}$. Since the robots travel alternately (odd and even robots), after two $round$s both the odd and the even robots proceed. We would like to show that D_p decreases by at least $2\rho(x)$ every 2 $round$s. We define the vertices of D_p as v_i and v_j (which reside on e_i and e_j respectively) - the two most distant points of the polygon. Define $v_{i'}$ as the intersection point between D_p and the edge after the proceeding of $\rho(x)$. Also define A as the intersection between edge e_i and the perpendicular line from $v_{i'}$ to e_i. Since $|\overline{(A, v_{i'})}| = \rho(x)$ and that the hypotenuse $(v_i, v_{i'})$ is bigger than the other edges in the right triangle $Av_iv_{i'}$, we get that $\overline{(v_i, v_{i'})} > \overline{(A, v_{i'})} = \rho(x)$. This is the same case for v_j, hence the distance between the two most distant edges of the polygon decrease by at least $2\rho(x)$. In the next time step the `fov`-overlap either stays the same (along the edges of the polygon) or increases (close to the vertices), thus the progress of the robots will be at least as the initial $\rho(x)$, since $\rho(x)$ is monotonically increasing. Therefore, if the diameter of the polygon P is D_p, the robots will necessarily meet after at most $\frac{2D_p}{2\rho(x)} = \frac{D_p}{\rho(x)}$ $round$s. Because each $round = \frac{\rho(x)}{v} + w$ time units, we get that the total runtime of the algorithm is at most $\frac{D_p}{\rho(x)} * (\frac{\rho(x)}{v} + w) = \frac{D_p * \rho(x)}{\rho(x)v} + \frac{D_p * w}{\rho(x)} = \frac{D_p}{v} + \frac{D_p * w}{\rho(x)} = \frac{D_p}{v} + \frac{D_p * w}{2\sqrt{dx - \frac{x^2}{4}}}$ □

Note that algorithm `ReduceArea` does not make use of the redundant robots, that is, a redundant robot is removed from the task. However, a redundant robot may potentially move inside the polygon and assist in disabling the locust agents. We leave this direction for future work.

V. Imperfect Enclosure

In the previous section we have examined the case of perfect enclosure, in which the locust agents are dominated by the robots and can not escape from the initial containing polygon, that is, $n \geq n^* = rmin(P)$. However, in many cases there are not enough resources to guarantee a perfect enclosure. We therefore examine in this section the case in which we have less than n^* robots to contain the locust, referred to as *Imperfect Enclosure*.

In the case of imperfect enclosure, the goal is to provide heuristics for the robots such that the percentage of locust individuals that are not eliminated by the robots (referred to as the *escaping locust*), is minimized.

Following the previous section, given enough robots, forming them around a closed polygon guarantees perfect enclosure for the locust agents within that polygon. Thus when given $n < n*$ robots, we examine different algorithms for converging to a closed polygon, for maximizing the number of eliminated locust agents. We define that polygon to be the `MEP` surrounding the centroid of the initial locations of locust agents such that the n robots can create a perfect enclosure along this polygon, and each robot is defined a *destination point* it needs to reach along this polygon. We have examined two convergence algorithms: the straight and spiral algorithms, described as follows. See illustration in Fig. 3. The *Straight Convergence Algorithm* is a simple algorithm in which the robots converge straightly to their destination point, as defined above. The time complexity of the algorithm is $\mathcal{O}(1)$ per robot.

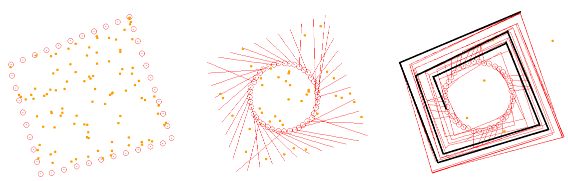

Fig. 3: The trails of the robots from their initial location (left) to their destination points in a straight movement (middle), and in a spiral movement (right). The black line emphasizes a trail of one specific robot.

The second algorithm is the *Spiral Movement Algorithm*. This algorithm is focusing on disablement of as many locust agents as possible. The spiral movement is predefined for each robot, and it travels along a "shrinking" polygon, each time by a distance d, until reaching the destination point. The time complexity of the algorithm per robot is $\mathcal{O}(|V| * \lceil \frac{distanceToCentroid}{2d} \rceil)$, where $|V|$ is the number of vertices and $distanceToCentroid$ is the distance between the centroid and the closest vertex to it.

A. Imperfect Enclosure - Evaluation

We have implemented both algorithms in the $2D$ TeamBots simulator[1], a simulator designed to support robot control systems and execution of teams of multiple robots. We describe here rigorous experiments we have conducted, focusing on the percentage of locust agents that were not eliminated (referred to as *escaping locust*). Due to space limits, we describe here a subset of the results.

[1]`https://www.cs.cmu.edu/~trb/TeamBots`

We have examined the two algorithms on 10 different worlds, using varying number of robots $(10, 20, 30, 40, 50)$, a fixed number (100) of randomly-moving locust agents, and different locust velocity compared to the robots' velocity (robot velocity was fixed to 8, and the locust velocity varied as $2, 4, 6, 8$). We wanted to examine the impact of the number of robots and locust velocity on the number of escaping locust. We have conducted a total of 8000 runs.

The left chart in Fig. 4 presents a correlation between the number of robots and the number of escaping locust. The more robots there are, more locust agents are being detected and disabled by the robots. Moreover, the spiral algorithm results in lower percentage of escaped locust agents compared to the straight convergence algorithm. The right chart shows a correlation between the spiral algorithm, locust velocity and the number of escaping locust: as the velocity of the locust agents increases, the number of escaping agents increases. However, for the straight convergence algorithm the difference is negligible: moving faster does not result in higher escape rate, as the straight algorithm does not focus on disablement of most of the locust but getting to the MEP as fast as possible.

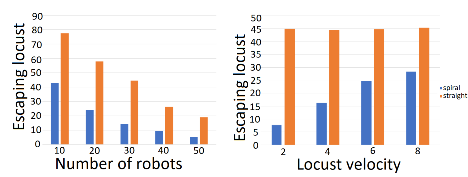

Fig. 4: On the left: the number of escaping locust with different number of enclosing robots. On the right: the number of escaping locust with different locust velocity.

Fig. 5 shows the correlation between the polygon area and the escaping locust. We can see that when using the straight algorithm, bigger area causes more escaping locust. However for the spiral algorithm, no significant difference was viewed as the polygon area grows.

VI. Conclusions and Future Work

We have presented the (CAD) problem, in which a team of robots are required to be placed around

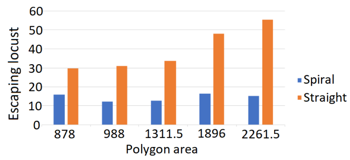

Fig. 5: The number of escaping locust in areas of different size. The x axis describes the polygon area size, the y axis describes the number of escaping locust.

a group of locust agents such that they fully contain them (*perfect enclosure*), and gradually decrease the enclosed area while disabling the locust agents. We examined the minimal number of robots guaranteeing perfect enclosure, and their behavior for solving CAD. We then examined empirically two possible behaviors for *imperfect enclosure*, in which there are not enough robots to guarantee perfect enclosure. There are still many directions to pursue in the future, among those making better use of the redundant robots in the `ReduceArea` algorithm, examining theoretically the case of imperfect enclosure, and evaluating more possible behaviors in this case.

REFERENCES

[1] D. Albani, J. IJsselmuiden, R. Haken, and V. Trianni, "Monitoring and mapping with robot swarms for agricultural applications," in *International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, 2017.

[2] D. Ball, P. Ross, A. English, T. Patten, B. Upcroft, R. Fitch, S. Sukkarieh, G. Wyeth, and P. Corke, "Robotics for sustainable broad-acre agriculture," in *Field and Service Robotics*, 2015.

[3] C. Zhang and N. Noguchi, "Development of a multi-robot tractor system for agriculture field work," *Computers and Electronics in Agriculture*, vol. 142, 2017.

[4] P. Gonzalez-de Santos, A. Ribeiro, C. Fernandez-Quintanilla, F. Lopez-Granados, M. Brandstoetter, S. Tomic, S. Pedrazzi, A. Peruzzi, G. Pajares, G. Kaplanis *et al.*, "Fleets of robots for environmentally-safe pest control in agriculture," *Precision Agriculture*, vol. 18, no. 4, 2017.

[5] Y. Elmaliach, N. Agmon, and G. A. Kaminka, "Multi-robot area patrol under frequency constraints," *Annals of Math and AI*, vol. 57, no. 3–4, 2010.

[6] J. J. Acevedo, B. C. Arrue, I. Maza, and A. Ollero, "Distributed approach for coverage and patrolling missions with a team of heterogeneous aerial robots under communication constraints," *International Journal of Advanced Robotic Systems*, vol. 10, no. 1, p. 28, 2013.

[7] E. Galceran and M. Carreras, "A survey on coverage path planning for robotics," *Robotics and Autonomous systems*, vol. 61, no. 12, 2013.

[8] K. J. Obermeye, A. Ganguli, and F. Bullo, "Multi-agent deployment for visibility coverage in polygonal environments with holes," *robust and nonlinear control*, 2011.

[9] Y. Shi, R. Li, and K. L. Teo, "Rotary enclosing control of second-order multi-agent systems for a group of targets," *International Journal of Systems Science*, vol. 48, no. 1, 2017.

[10] J. Guo, G. Yan, and Z. Lin, "Local control strategy for moving-target-enclosing under dynamically changing network topology," *Systems and Control Letters*, vol. 59, no. 10, 2010.

[11] M. Kubo, A. Yamaguchi, T. Yoshimura, and H. Sato, "Swarm based enclosure model for unspecified number of targets," in *IEEE/SICE International Symposium on System Integration (SII)*, 2011.

[12] H. Park and S. Hutchinson, "A distributed robust convergence algorithm for multi-robot systems in the presence of faulty robots," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015.

[13] J.-M. Lien, S. Rodriguez, J.-P. Malric, and N. M. Amato, "Shepherding behaviors with multiple shepherds," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2005.

[14] A. Pierson and M. Schwager, "Bio-inspired non-cooperative multi-robot herding," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2015.

[15] P. I. Cowling and C. Gmeinwieser, "Ai for herding sheep," in *Sixth Artificial Intelligence and Interactive Digital Entertainment Conference*, 2010.

[16] M. Evered, P. Burling, M. Trotter *et al.*, "An investigation of predator response in robotic herding of sheep," *International Proceedings of Chemical, Biological and Environmental Engineering*, vol. 63, 2014.

[17] L. Cheng, Y. Wang, W. Ren, Z.-G. Hou, and M. Tan, "Containment control of multiagent systems with dynamic leaders based on a pi^n-type approach," *IEEE transactions on cybernetics*, vol. 46, no. 12, 2016.

[18] B. U. Mehendale, "Potential field based approach for multi-threat containment with cooperative robots," Ph.D. dissertation, Rochester Institute of Technology, 2006.

[19] M. Kubo, H. Sato, A. Yamaguchi, E. Kitamura, and A. Namatame, "Probability of mixing up the nearest neighbor robot under target enclosure by robot swarm," *Journal of Robotics, Networking and Artificial Life*, vol. 2, no. 3, 2015.

[20] K. Genter, N. Agmon, and P. Stone, "Ad hoc teamwork for leading a flock," in *International Conference on Autonomous Agents and Multiagent Systems*, 2013.

[21] P. Stone, G. A. Kaminka, S. Kraus, and J. Rosenschein, "Ad hoc autonomous agent teams: Collaboration without pre-coordination." in *AAAI*, 2010.

[22] L. Huang, M. Zhou, K. Hao, and E. Hou, "A survey of multi-robot regular and adversarial patrolling," *IEEE/CAA JOURNAL OF AUTOMATICA SINICA*, vol. 6, no. 4, 2019.

[23] G. Rabanca and I. Vigan, "Covering the boundary of a simple polygon with geodesic unit disks," *arXiv preprint arXiv:1407.0614*, 2014.