Patrolling and target tracking using cooperative artificial agents

Pedro Carrasqueira
pedro.carrasqueira@ist.utl.pt

Instituto Superior Técnico, Lisboa, Portugal

October 2018

Abstract

We present a novel extension for Loyd's k-means algorithm, that transforms the minimization of the distance to points into distance to balls, to solve three different target management missions. Using an Anchor Point discretization of the region of interest (ROI) via a stochastic model, we formulate an initial deployment for a known number of sensors. The concept of masses attributed to each individual Anchor Point, with their constant update, gives us the necessary tool to Patrol the ROI or Track and Escort targets. The formulation takes into consideration the sensors coverage range and speed limitations where different sensors with different characteristics might be employed. Once the Anchor Points are defined or a Target identified the algorithm runs autonomously in a robust, cooperative and adaptable formulation that uses a minimization based approach that evidences excellent coverage rates.
Keywords: k-means, Area Coverage, Patrolling, Target Tracking and Escorting, mobile agent network

I. Introduction

Autonomous patrolling is becoming a subject of great practical and scientific importance. From oceanic and land surveillance to patient monitoring or even simple waiter robots, examples abound. However, the limits imposed by the sensors technology, either in velocity or range, make the problem of finding an optimal trajectory using the least amount of network resources very difficult.

In small areas the number of necessary sensors will never be prohibitive, thus a low cost solution could be easy to find. The necessity of covering large areas, on the other hand, implies a large number of sensor nodes. Mobile sensors reduce the costs by being able to cover different areas as time evolves therefore reducing the number of necessary sensors. The employment of a cooperative covering strategy with these sensors is not trivial and usually leads to greedy methods. The best route that each particular sensor should take in order to achieve a good coverage is of great importance in order to achieve as small costs as possible.

In areas with high sensor density is relatively easy to find acceptable solutions when it comes to tackling the problem of efficiently covering an area. Finding a solution for small density where some of them keep track of a target and the remaining area is still covered by the team of agents is much more difficult and usually leads to greedy methods where a global minimum is not taken into account.

A. Relevant work

The field of multi-robot target detection and tracking as defined by [11] usually considers three types of robots/sensors/agents: stationary [14, 7], mobile [3, 9, 10] or a mix of mobile and stationary [6]. The employment of stationary sensors in large regions leads to a high number of necessary sensors and therefore to prohibitive costs.

To achieve the best coverage of the area the works cited either deal with a representative grid of the area [6], a point discretization [14, 13] or area partitions [7]. Each of the approaches has their advantages and disadvantages. However, in [14], a point coverage problem aiming at cost minimization of a Wireless Sensor Network (WSN) with limited range sensors is solved. The objective is to find a set from all the sensors that achieves coverage of an area with some probability at each step with the lowest cost. The implemented division of the region of interest (ROI) allows to divide it in a finite set of points that, if all covered, achieve coverage with the necessary probability. However, they only solve the problem with static and high sensor density areas, therefore, its scope is limited.

In [7] area partitions are considered where the main objective is to find the deployment of a set of sensors such that the area is covered by at least K sensors, *i.e.*, K-coverage. Their formulation is non-convex; however, the authors proposed an algorithm that converges to a local minimum of the problem. For their approach, the area is divided into partitions according to the concept of K-order Voronoi diagram. Each partition has K sensors responsible for it and the sensors positions are the centroids over all the partitions they are responsible for. They are able to deploy sensors in 2D areas with obstacles and 3D surfaces, however the solution for sensors whith limited range is to employ more sensors until the sensors attain K-coverage. This is impractical in low sensor density situations where we want to minimize the number of sensors and use them to their full potential.

The use of a mix between mobile and static sensors allows to complement data taken by the static sensors with the data from the mobile ones. In [6] the area is divided in a grid where mobile sensors are forced to always be moving and have movement and sensing restrictions. The next position is found taken in consideration a discrete set of possibilities and a set of functions who aim at representing different characteristics of the environment. A weighted sum of the functions values is made according to the mission objectives. This solution is also applicable for the case without static sensor nodes and is able to take into consideration limits in range and velocity. However, it relays on a greedy solution since each of the mobile sensors takes only into consideration what is best for it and a global optimum is not the individual objective, leading to a nonoptimal use of resources.

When mobile sensors are considered the main challenge is to define the path for each sensor so that all the interest area is covered, ideally, the same number of times. The most obvious solution is to define a cyclic route such that every spot is visited the same number of times. This solution is deterministic and makes it easy for an intruder to find a perfect time for attack, however, it is able to cover every area partition. Other approaches use probabilities to define their patrolling policy. Usually, two approaches are found: Markov Chains [3] or Stackelberg Games [9]. The definition of probabilities results in an uncertainty that makes it difficult to predict their path, however never ensures that all the area is visited.

To track movable intruders an approach is developed in [10] where the problem of covering an area and simultaneously escort a target is addressed using Voronoi partitions of the area. The authors use density functions to represent the Voronoi partition and define the best positions of the sensors if they have targets to cover. Whenever the targets position changes so will the density function, making the robot move in a trajectory similar to the target while not forgetting the remaining area of its Voronoi partition. The robots that do not have a target to escort will stay in the centroid of their respective Voronoi partitions. The employment of the density function is therefore limited and could be used to define trajectories for the remaining robots. This approach was verified in real robot situations.

B. Contributions

In this paper we give solution to three different missions using a common background. Our contributions can be summarized as:

- We formalize a novel extension for the k-means problem, that we term k-disks, which takes into consideration the distance to balls instead of the distance to points and that is used as background for solving three different missions, where the agents are not explicitly commanded to enter in different states: Area Coverage, Patrolling and Target Tracking and Escorting .
- Using a probabilistic model for the sensors we define a non-symmetric Anchor Point placement strategy where every point gives information about the coverage probability of its surrounding area.
- Using the Anchor Point definition and the novel k-disks approach we propose a solution that works in any dimension and takes into consideration both sensing range and velocity limitations.
- We formulate an approach that is able to efficiently perform tracking of a target while the remaining network performs Patrolling of the remaining region of interest (ROI), without explicitly instructing which agent does what. Each agent's role will be a consequence of an overall cost function.
- "k-disk: placement, patrolling and tracking for a team of mobile agents" in preparation, to be submitted to the IEEE Transactions on Control Systems and Technology

II. Problem Formulation

We will consider a set of sensors $\mathcal{S}$ deployed in a ROI where s_j stands for the j^{th} sensors position. Every sensor s_j has a limited range R_j for which the detection of an event is ensured. For distances higher than R_j we use a stochastic sensor model that has been widely used in [14, 1, 15]. Defining W_{ij} as: point p_i is covered by sensor $s_j \in \mathcal{S}$. The model is explicitly defined by

$$p(w_{ij}|d_{ji}) = e^{-k_j d_{ji}}, \tag{1}$$

where k_j is a positive constant that varies with the type of sensors considered and d_{ji} is the distance between the arbitrary point p_i and a ball centered in sensor s_j with radius R_j.

The advantage of this model is that it has probabilities decreasing with distance making it more realistic and since it is stochastic takes into consideration unknown environmental issues such as obstacles. The value of parameter k_j gives the model the freedom to adapt to any environment and any type of sensors employed in a real situation.

From [14] we adopt the ε-full area coverage definition.

Definition 1 (ε-full area coverage): Sensor set $\mathcal{S}$ provides ε-full area coverage to the ROI if coverage of any point in ROI is no less than ε, *i.e.*

$$\forall p_i \in \text{ROI}, \quad \sum_{j \in \mathcal{S}} p(w_{ij}|d_{ji}) \geq \varepsilon. \tag{2}$$

In our case the goal is to obtain the positions of $\mathcal{S}$ such that ε-full area coverage is obtained or at least the largest area of the ROI is covered. As one can imagine such a problem is quite complex and is only solvable if the area is divided in a grid or a finite set of points. Some references found in the literature [6, 1, 15, **?**] have used point discretization; however, the division is made without criteria. In [14] a different approach is used. Using model (1) it is possible to define a set of Anchor Points for which if an Anchor Point is covered with some probability $\tau > \varepsilon$ then a ball with radius b can be imagined around that point such that every point inside the ball has covering probability greater than ε. The value of b is obtained by

$$\tau e^{-kb} = \varepsilon \Leftrightarrow b = -\frac{1}{k} \log\left(\frac{\varepsilon}{\tau}\right). \tag{3}$$

After obtaining the distance b, the set of Anchor Points $\mathcal{A}$ can be defined in such way that all points of $\mathcal{A}$ have neighbour points at distance smaller than $2b$ and the union of all balls around each point $a_k \in \mathcal{A}$ contains the ROI. This can be written as minimization problem

$$\begin{aligned} \min \quad & |\mathcal{A}| \\ \text{s.t.} \quad & \|a_k - a_l\| \leq 2b, \quad \forall a_k \in \mathcal{A}, \quad a_l \in \mathcal{N}(a_k) \\ & \text{ROI} \subset \bigcup_{a_k \in \mathcal{A}} B_{a_k}(x) \end{aligned} \tag{4}$$

where $|\mathcal{A}|$ is the cardinality of set $\mathcal{A}$, $\mathcal{N}(a_k)$ is the set of first neighbours of a_k and $B_{a_k}(x)$ is a ball surrounding a_k with radius b. Solving this problem is very difficult because it is minimizing a set cardinality.

It is a necessary condition for this point distribution that $\tau > \varepsilon$ otherwise $b = 0$ and no Anchor Points will be found due to the definition (3). It is also of interest to note that, as the value of $\frac{\varepsilon}{\tau}$ decreases, the value of b increases thus simplifying the problem because the number of necessary Anchor Points $|\mathcal{A}|$ theoretically decreases.

As stated in [14] this approach transforms an area coverage problem into a point coverage problem where by covering a certain point with probability τ we can assure that a ball of radius b around that point is covered with probability ε.

In this article we will only present the algorithm for Anchor Point placement in a 2D ROI with known width and height.

A. k-disks: k-means extension

The most used formulation for clustering is k-means which has the objective of grouping a set of points of any dimension into k clusters such that distances between any point in a cluster and the centroid of the cluster is minimized. Considering a set of points X that will be clustered with $x_i \in X$ and $\mathcal{S}$ the set of centroids such that $s_i \in \mathcal{S}$ the problem is formulated as

$$\min_{\mathcal{S}} \quad \sum_{i=1}^{X} \min\left(\|x_i - s_1\|_2^2, \|x_i - s_2\|_2^2, ..., \|x_i - s_k\|_2^2\right). \tag{5}$$

For $k > 1$, (5) is a combinatorial problem and therefore NP-hard as proven in literature [4]. Even though usually non-convex and difficult to solve there is a large variety of algorithms that give good approximations to solve the problem from which the most widely used is Lloyd's algorithm [8] together with k-means++ [2], a probabilistic scheme to initialize k-means.

A solution that uses the standard k-means formulation would come with the disadvantage of not considering the range of the sensors to find the optimal positions. Our proposed extension is therefore to change the minimization problem in order to consider the range limitations:

$$\min_{\mathcal{S}} \quad \sum_{i=1}^{|\mathcal{A}|} \frac{w_i}{2} \left(\min\left(d_{i1}(s_1)^2, d_{i2}(s_2)^2, ..., d_{ik}(s_k)^2\right)\right), \tag{6}$$

where w_i is a mass that is constant to each ball during the resolution of the problem, whose values will depend on the mission considered and

$$d_{ij} = (\|a_i - s_j\| - R_{j\tau})_+ \tag{7}$$

is the distance from sensor s_j to a ball centered in Anchor Point a_i with radius $R_{j\tau}$ which corresponds to the distance for which an event has detection probability τ by sensor s_j and where $(x)_+ = \max\{0, x\}$.

Problem (6) transforms the usual k-means problem that minimizes the distance to points into k-disks, that minimizes the distance to balls, allowing us to take into consideration the sensing range of each individual sensor.

Since this is still a combinatorial problem it is also very difficult to find a solution. To solve it we developed an adaptation of both Lloyd's algorithm and k-means++. First we used k-means++ exactly like defined in [2] to define the initial position of set $\mathcal{S}$. Once they are defined we used the concept of alternate minimization defined by Lloyd in [8]. We first divide the set of Anchor Points $\mathcal{A}$ according to the distances to balls centered in the sensors $\mathcal{S}$ with radius $R_{j\tau}$ in order to have $|\mathcal{S}|$ clusters $c_j \in C$ such that $\bigcup_{j=1}^{|C|} c_j = \mathcal{A}$.

The initial division in the clusters c_j allows to have $|\mathcal{S}|$ minimization problems of the form

$$\min_{s_j} \quad \sum_{i=1}^{|c_j|} \frac{w_i}{2} \left(d_{ij}(s_j)\right)^2, \tag{8}$$

for a general s_j of $\mathcal{S}$.

Problem (8) is convex has proven in the thesis, so it has a unique solution, possible to find in polynomial time. The alternated minimization is performed until the sensors positions converge and do not change.

In the Area Coverage mission we want to find the initial position that minimizes the sum of distances of the Anchor Points to the sensors so that the minimization problem we want to solve is exactly (8). For the Patrolling and Target Tracking and Escorting mission we also take into consideration the limitations in the movement of the sensor, so we will add a constraint to problem (8). For this missions we will have to solve problem:

$$\begin{aligned} \min_{s_j(t)} \quad & \sum_{i=1}^{|c_j|} \frac{w_i}{2} \left(d_{ij}(s_j(t))\right)^2, \\ \text{s.t.} \quad & \|s_j(t) - s_j(t-1)\| \leq d_j \end{aligned} \tag{9}$$

where, besides the temporal consideration given by t, we also add a constrain that connects two consecutive time steps fixing a limit on the movement of the sensor d_j for the time difference considered.

We already know that the objective function is convex, however for (9) to be a convex problem the constraint also has to be convex. Since the constrain is a norm, which is assured to be convex due to the triangle inequality, we know that the constrain is convex, as well as, problem (9).

B. Mass coefficients w_i

The mass coefficients are defined according to the mission we want to accomplish. For the Area Coverage mission, since we want to find the best overall position for the set $\mathcal{S}$, where all balls are equally important they will all have the same mass $w_i = 1$. It is possible to consider from the outset the existance of balls that are more important to cover than others by considering higher values for the masses. However, this is not the main goal.

For the Patrolling mission, the first concern in the definition of the masses is that no points should be forgotten. Defining Δ_i as the difference between the current time step and the last time the ball i was scanned, the masses have to take into consideration a temporal exploration such that the largest the value of Δ_i, the larger the w_i, and the more likely this point is to be covered in the next iteration.

If we only take into consideration the temporal exploration we will have all masses of not scanned points in the same cluster c_j equal. Since the scanned points will always be the same and the distances from the sensors to the scanned points are always 0 we would get into a static situation because the two problems for Area Coverage and Patrolling

$$\begin{aligned} &\underset{s_j}{\text{minimize}} \quad \sum_{i=1}^{|c_j|} \frac{1}{2} \left(d_{ij}(s_j)\right)^2 \Leftrightarrow \\ \Leftrightarrow\ &\underset{s_j}{\text{minimize}} \quad w \sum_{i=1}^{|c_j|} \frac{1}{2} \left(d_{ij}(s_j)\right)^2, \end{aligned} \tag{10}$$

would be returning the same minimizer.

Since the distances between points and the sensors will be different as long as there are no symmetries. The masses will be defined as

$$w_i(t) = \alpha \frac{\Delta_i^m}{\sum_{k=1}^{|c_j|} \Delta_k^m} + (1-\alpha) \frac{d_{ij}(s_j(t-1))^n}{\sum_{k=1}^{|c_j|} d_{kj}(s_j(t-1))^n}. \tag{11}$$

The value of α is in the interval $0 \leq \alpha \leq 1$ and should be chosen taking into consideration a compromise between time and spatial exploration. A few comments should be made:

- The weights $w_i(t)$ are normalized to 1 for each cluster of points c_j.
- The value of $m > 0$ determines the importance of an increment of the time factor.
- The value of $n > 0$ determines the importance of an increment of the spatial factor.
- If $\alpha = 0$ only the spatial factor will be considered which does not guarantee that every point will be visited.
- If $\alpha = 1$ only the time factor is considered so we will return to the impractical static case.

In a ROI where there is lack of resources there are areas that will not be covered by the initial deployment of the Area Coverage mission. In a symmetric situation the sensors lean to the centroid of the points thus they are all at the same distance. In such situations, by looking at the mass definition (11), it is easy to see that in cases where the Anchor

Points a sensor is in charge of covering are symmetric the weighted sum of the spatial and temporal factors will be equal for layers of points at the same distances. Therefore, the problem will be equivalent to the Area Coverage mission yielding a constant optimal position for the sensor. To avoid such effects, non-symmetric point placement is needed.

In the Target Tracking and Escorting mission the objective is to do the Patrolling of the ROI and at the same time follow the target. Two sets will be considered, the already defined set of Anchor Points $\mathcal{A}$ and the intruder set $\mathcal{I}$. The division of the sets in Anchor Points and intruders is important to define the masses which will be given by

$$w_i(t) = \begin{cases} (11), & \text{if } c_{ji} \in \mathcal{A} \\ N, & \text{if } c_{ji} \in \mathcal{I} \end{cases}. \tag{12}$$

where c_{ji} stands for the i element of the set c_j and the normalization of (11) only takes in consideration the Anchor Points A.

The principle behind (12) is to use the same mass definition for the Anchor Points that was used in Patrolling so that the sensors are able to continuously perform the Patrolling mission. The mass for the intruders will be a constant given by $N \gg 1$. The rationale for this number is as follows: the masses have to evidence the importance of the point, so the intruders need a mass higher than the Anchor Points. Since the the Anchor Points masses are always normalized to 1, to have a higher value for the intruders, it is enough to have $N \gg 1$.

III. Algorithm derivation

To solve our problem the first step is to find a solution of problem (4) that is not symmetric. Once the Anchor Points are defined we can use our k-disks, an adaptation of Lloyd's algorithm, in association with k-means++ to find the solution to the three missions.

A. Point Placement

Considering a rectangular ROI the most obvious way to solve problem (4) is by defining points in a grid such that each vertex of the grid is an Anchor Point. The grid will be composed of squares with diagonal $2b$ in such way that the side of each square will be given by

$$l = \sqrt{2}b. \tag{13}$$

It is straightforward to see that the definition in a grid would be symmetric. This placement is a solution of problem (4). However, there are an infinite number of other solutions that are not symmetric.

A different, non-symmetric, definition of points is therefore used. We know that the distances between vertices of the symmetric case are all equal, so instead of defining all distances equal to l we can use random number generation to define different distances smaller than l between consecutive Anchor Points so that the placement is non-symmetric but still feasible and solution of (4). For this new definition we need to define a tolerance parameter γ such that every Anchor Point is at a distance between the maximum possible l and $l - \gamma$. The pseudo-code used for point placement is given in Algorithm 1.

Algorithm 1: Point placement for a rectangular ROI.

input : ROI width w, ROI height h,
Tolerance γ, Square size l.
output: Set with points distribution points.

begin
  $x_{\max} \leftarrow 0$;
  $y_{\max} \leftarrow 0$;
  first $\leftarrow 1$;
  first_x $\leftarrow 0$;
  while $x_{\max} -$ first_x $\leq w$ **do**
    column $\leftarrow \emptyset$;
    aux_w $\leftarrow x_{\max} - U(0, \gamma)$;
    aux_h $\leftarrow y_{\max} - U(0, \gamma)$;
    first_y $\leftarrow$aux_h;
    while first_y $-$ aux_h $\leq h$ **do**
      Concatenate [aux_w,aux_h] **to** column;
      $y_{\max} \leftarrow$ aux_h $+l$;
      aux_w $\leftarrow x_{\max} - U(0, \gamma)$;
      aux_h $\leftarrow y_{\max} - U(0, \gamma)$;
    end
    center column **with** h ;
    $y_{\max} \leftarrow l/2$;
    if first $== 1$ **then**
      first_x $\leftarrow$ minimum x of column;
      first $\leftarrow 0$;
    end
    $x_{\max} \leftarrow \min($first_x $+l, w-$ first_y$)$;
    add column **to** points;
  end
  center points **with** w
end

This point definition is just an example among an infinite number of possibilities. This algorithm can easily be extended to any D-dimension where the only difference will be that, besides $x_{\max}$ and $y_{\max}$, there would also be $D - 2$ parameters that would be updated with the same principle as $x_{\max}$.

Using this algorithm in the case where the value of the tolerance is $\gamma = 0$ we return to the unpractical case were the Anchor Points are symmetric.

Even though the algorithm is presented for a rectangular ROI, it is interesting to note that it can also be adapted for any other ROI. In that situation, one could consider the longest line inside the ROI as $a = \max\{\|x - y\|\}$, $\forall x, y \in$ ROI, and use Algorithm 1 considering an area $a \times a$ to get a set of Anchor Points from which only those inside the ROI are considered. The only requirement is to carefully deal with the borders in order to get a feasible distribution.

The higher the value of the tolerance γ the more spread the points are. The increment in the tolerance value as a general rule also leads to a higher number of Anchor Points and therefore an increment of the problem complexity for solving the three proposed missions. The tolerance parameter has to be chosen with a compromise between increasing the problem complexity and spreading the Anchor Points.

Algorithm 2: k-disks Patrol and Tracking.

```
   input : Anchor Point set points, Number of
           sensors n_sensors, Number of attempts
           attempts, Mass update parameters
           α, m, n, N.
   output: Sensor placement sensors at each
           time step.
1  begin
2      possibility ← k-means++ with (n_sensors,
        points);
3      calculate cost for possibility;
4      min ← cost;
5      sensors ← possibility;
6      for i = 2 to attempts do
7          possibility ← k-means++ with
            (n_sensors, points);
8          calculate cost for possibility;
9          if cost < min then
10             min ← cost;
11             sensors ← possibility;
12         end
13     end
14     do indefinitely
15         Update T with scanned path;
16         Update W with (12);
17         do Lloyd's k-means extension
            algorithm with (points,sensors,W)
            obtain sensors;
18     end
19 end
```

B. k-disks Patrolling and Tracking

Even though we proposed Algorithm 1 for the Anchor Point placement, it is not mandatory to use it since any other non-symmetric Anchor Point placement that is solution of (4) is also usable.

In Section II we formulated the problem with a k-means extension and adapted Lloyd's algorithm to solve it. In this section we proposed Algorithm 2 that works for any arbitrary Anchor Point set $\mathcal{A}$.

In Algorithm 2, W is defined as the set of masses composed by all w_i and T is the set with the time difference between the current iteration and the last iteration the Anchor Points were scanned, *i.e.*, Δ_i.

Steps 2 and 7 are the solutions of the k-means++ sensor placement followed by the k-disks adaptation to Lloyd's algorithm. Step 17 is the solution of the k-disks problem with the respective Anchor Point distribution and the constrained minimization problem (9).

Even though we have already presented the basic concepts behind k-disks algorithm we still have not described how we solved the convex minimization problems (8) and (9).

We have already proved that (8) is convex and that (9) is also convex with convex constraints. For (8) to be under the conditions to use Fast Iterative Shrinkage-Thresholding Algorithm (FISTA) [12] the objective function has to be convex, differentiable with domain $\mathbb{R}^n$. We already proved that it is convex and has domain $\mathbb{R}^n$ so we just have to find the gradient of the function.

From [5, Chapter X, Proposition 3.2.2 and Theorem 3.2.3] we know that

$$\nabla\left(\frac{1}{2}d_S^2(x)\right) = x - P_S(x), \tag{14}$$

where S is a convex set and $d_S(x)$ is the distance between set S and point x.

A ball is known to be a convex set so the gradient of $f(s_j) = \sum_{i=1}^{|c_j|} \frac{w_i}{2}\left(d_{ij}(s_j)\right)^2$ is given by

$$\begin{aligned}\nabla f(s_j) &= \nabla\left(\sum_{i=1}^{|c_j|} \frac{w_i}{2}\left(d_{ij}(s_j)\right)^2\right) \\ &= \sum_{i=1}^{|c_j|} w_i \, \nabla\left(\frac{1}{2}\left(d_{ij}(s_j)\right)^2\right). \\ &= \sum_{i=1}^{|c_j|} w_i \left(s_j - P_{B_{ij}}(s_j)\right)\end{aligned} \tag{15}$$

The gradient requires the expression of the projection into a ball which, for a generic ball B_c centered at c with radius R, is given by

$$P_{B_c}(x) = \begin{cases} x, & \text{if } \|x - c\| \leq R \\ c + R\frac{x-c}{\|x-c\|}, & \text{otherwise} \end{cases}. \tag{16}$$

This shows us that we are under all conditions of using FISTA to solve (8). Since (9) is equal to (8) with a constrain we just have to take into consideration the FISTA conditions for the constrain. The constrain is a limit on the movement of the sensor such that $s_j(t)$ can only be in a ball surrounding the previous position. Therefore, the constrain is a closed and convex set with a known prox operator given by the projection into the ball centered in $s_j(t-1)$ with radius d_j as given by (16). We are, therefore, in problem (9) also under all conditions to aply FISTA.

The apply FISTA we need to find a good step size. Finding a Lipschitz constant L for the gradient of the objective functions allows to know that the algorithm descents as fast as $\mathcal{O}(1/k^2)$ for a step $t_k = 1/L$. A function f is said to be Lipschitz continuous with parameter $L > 0$ if:

$$\|f(x) - f(y)\|_2 \leq L\|x - y\|_2, \quad \forall x, y \in \mathbf{dom}(f). \tag{17}$$

The value of the Lipschitz constant depend on the masses w_i and therefore on the mission considered. In the Area Coverage and Patrolling all masses are normalized or equal to 1 so we get $L = 2|c_j|$, where $|c_j|$ is the cardinality of the Anchor Point partition of s_j. For the Target Tracking and Escorting mission we get $L = 2|A_j| + 2N|I_j|$, where A_j is the set of Anchor Points s_j is in charge of covering, I_j is the set of intruders agent s_j is in charge of covering and N is the mass of an intruder as defined in (12).

We emphasize that the value of the Lipschitz constant L is not unique. In fact if l is proved to be the Lipshitz constant of f than any $L \geq l$ is also a possible Lipshitz constant. If we define $L = 2|A_j| + 2N|I_j|$ we will allow Algorithm 2 to run autonomously with the same Lipschitz constant for all missions and as a result without explicit determination of what mission each sensor is performing.

The gradient of the objective function of the two problems (8) and (9) is the same, so we will demonstrate the Lipschitz constants taking into consideration the differences in the masses definition. The gradient was found in (15) and is given for an arbitrary x by $\sum_{i=1}^{|c_j|} w_i \left(x - P_{B_{ij}}(x)\right)$. For notation simplicity and without loss of generality we will from now on represent $P_{B_{ij}}$ as P_{B_i}. To find L we start by writing the problem as

$$\begin{aligned}\|\sum_{i=1}^{|c_j|} w_i(x - P_{B_i}(x)) - \sum_{i=1}^{|c_j|} w_i(y - P_{B_i}(y))\| = \\ = \|\sum_{i=1}^{|c_j|} w_i(x - y) + \sum_{i=1}^{|c_j|} w_i(P_{B_i}(y) - P_{B_i}(x))\|,\end{aligned} \tag{18}$$

where x and y are any arbitrary points in the domain of the objective function. Using the triangle inequality we arrive at

$$\begin{aligned}\|\sum_{i=1}^{|c_j|} w_i(x - y) + \sum_{i=1}^{|c_j|} w_i(P_{B_i}(y) - P_{B_i}(x))\| \leq \\ \leq \|\sum_{i=1}^{|c_j|} w_i(x - y)\| + \|\sum_{i=1}^{|c_j|} w_i(P_{B_i}(y) - P_{B_i}(x))\|.\end{aligned} \tag{19}$$

For any set of N function $f_i(x)$ we have

$$\|\sum_{i=1}^{N} f_i(x)\| \leq \sum_{i=1}^{N} \|f_i(x)\| \tag{20}$$

as can be proven by applying consecutive triangle inequalities between the sums of all members of the set. Applying this result to our specific case and since the masses w_i are positive constants we can write (19) as

$$\begin{aligned}\|\sum_{i=1}^{|c_j|} w_i(x - y)\| + \|\sum_{i=1}^{|c_j|} w_i(P_{B_i}(y) - P_{B_i}(x))\| \leq \\ \leq \sum_{i=1}^{|c_j|} w_i\|x - y\| + \sum_{i=1}^{|c_j|} w_i\|P_{B_i}(y) - P_{B_i}(x)\|.\end{aligned} \tag{21}$$

For the Area Coverage mission we have $w_i = 1$ and for the Patrolling mission we have $w_i \leq 1$, therefore, for these two missions, we get

$$\begin{aligned}\sum_{i=1}^{|c_j|} w_i\|x - y\| + \sum_{i=1}^{|c_j|} w_i\|P_{B_i}(y) - P_{B_i}(x)\| \leq \\ \leq \sum_{i=1}^{|c_j|} \|x - y\| + \sum_{i=1}^{|c_j|} \|P_{B_i}(y) - P_{B_i}(x)\|.\end{aligned} \tag{22}$$

It is known that for every closed convex set $\mathcal{S}$ and for any points $x, y \in \mathbb{R}^N$ we have $\|P_{\mathcal{S}}(x) - P_{\mathcal{S}}(y)\| \leq \|x - y\|$ because there are only a few possibilities:

- Points $x, y \in \mathcal{S}$ and therefore $P_{\mathcal{S}}(x) = x$ and $P_{\mathcal{S}}(y) = y$, so $\|P_{\mathcal{S}}(x) - P_{\mathcal{S}}(y)\| = \|x - y\|$.
- Point $x \in \mathcal{S}$ and $y \notin \mathcal{S}$ so that $\|P_{\mathcal{S}}(x) - P_{\mathcal{S}}(y)\| < \|x - y\|$.
- Points $x, y \notin \mathcal{S}$ which results in $\|P_{\mathcal{S}}(x) - P_{\mathcal{S}}(y)\| \leq \|x - y\|$, as given in [5].

Thus, it is said that the projection is a contraction operator. Using this property and since a ball is a closed convex set we get

$$\begin{aligned}\sum_{i=1}^{|c_j|} \|x - y\| + \sum_{i=1}^{|c_j|} \|P_{B_i}(y) - P_{B_i}(x)\| \leq \\ \leq \sum_{i=1}^{|c_j|} \|x - y\| + \sum_{i=1}^{|c_j|} \|y - x\|,\end{aligned} \tag{23}$$

and with some algebraic manipulation we arrive at

$$\sum_{i=1}^{|c_j|}\|x-y\|+\sum_{i=1}^{|c_j|}\|y-x\| = 2\sum_{i=1}^{|c_j|}\|x-y\| = 2|c_j|\|x-y\|, \tag{24}$$

which yields the final value for the Area Coverage and Patrolling missions of

$$L = 2|c_j|. \tag{25}$$

For the Target Tracking and Escorting mission we split set c_j in the Anchor Points A and the intruders I, such that $c_j = A \bigcup I$. To proof the Lipschitz constant value we use the triangle inequality to divide the demonstration in two problems according to the points taken into consideration. One demonstration for the set A and the other for I.

Because the masses for the Anchor Points in the Target Tracking and Escorting mission match those in the Patrolling mission we get the same Lipschitz constant of $L_A = 2|A|$ for set A. For the intruders set I the proof starts in the same step as (21) where instead of c_j the sum is in I. Using the fact that for the intruders $w_i = N$ with the same reasoning already done for the previous missions we get $L_I = 2N|I|$. Gathering the results for each of the sets A and I we get the final result

$$L = L_A + L_i = 2|A| + 2N|I|. \tag{26}$$

IV. Results

The simulation results presented in this section were obtained considering a squared ROI with 100m of width w and height h where, for the Anchor Point placement, we used tight limits for the covering probabilities of $\varepsilon = 0.8$, $\tau = 1$ and a tolerance of $\gamma = \frac{l}{4}$ for the placements obtained by Algorithm 1. In the ROI, we will deploy 10 sensors with equal characteristics even though Algorithm 2 works for different sensors. We will consider $R_j = 5$m and $k_j = 0.05$ for the stochastic model presented in (1), with a speed constraint of $d_j = 10$m. We ran k-means++ 30 times until an initial deployment is defined and we used $n = 50$, $m = 75$, $\alpha = 0.7$ and $N = 1000$ to perform the Patrolling, and Target Tracking and Escorting missions.

A. Area Coverage

The Area Coverage mission takes into consideration a static situation that is between lines 2 and 13 of Algorithm 2. The main goal is to define the initial position such that the ROI is better covered, to attain this goal, we take into consideration the objective functions values after the convergence of k-means++ and k-disks. From all k-means++ trials we chose the one with the lowest objective function value. This criteria ensures that the chosen placement is the one that minimizes the distance between the balls centered in the sensors and the Anchor Points, therefore, assuring that there is higher probability of detecting an intruder in the ROI.

From the start, it is obvious that the 10 sensors with the defined characteristics are not able to achieve ε-full area coverage since the sum of their coverage balls with radius $R_{j\varepsilon}$ is smaller than the total area. Even though ε-full area coverage is not achieved there are solutions for the Area Coverage mission that are better than others since k-means++ is a randomized algorithm. For the considered ROI and sensor characteristics considered in the same 30 runs we get objective function values of (8) that go from 2.92×10^4 until as much as 3.59×10^4. The differences arise mainly in the number of Anchor Points that each sensor is in charge of covering, the more equally distributed they are the smaller value is achieved yielding better overall coverage. The instance corresponding to the objective function value of 2.92×10^4 is shown in Figure 1.

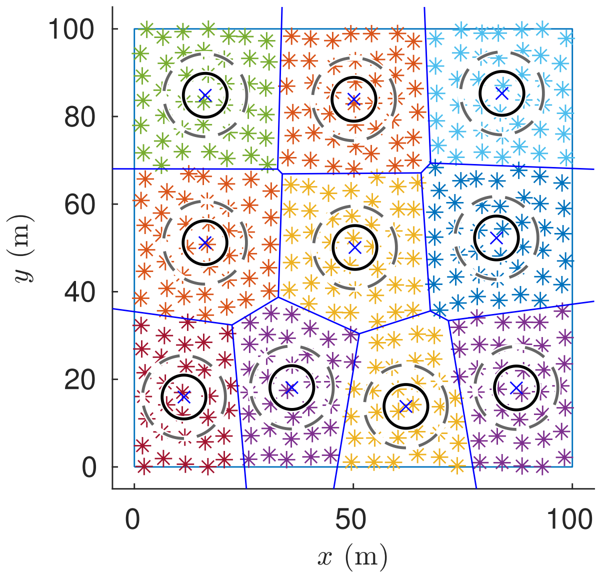

Figure 1: The number of Anchor Points of each of the partitions is similar for all of them and therefore a better overall coverage is achieved. The crosses represent the sensors, the asterisk the Anchor Points where the different colors represent the different sensors distribution, the lines represent Voronoi cells for each sensor, the solid circle $R_{j\tau}$ (radius for which an event is detected with probability of τ) and the dashed circle $R_{j\varepsilon}$ (radius for which an event is detected with probability of ε).

The distribution of the Anchor Points for the different sensors are represented in different colors and divided by Voronoi cells. It is interesting to note that at the algorithm convergence all Anchor Points that have the same color are also inside the same Voronoi cell. Another interesting characteristic is that the number of Anchor Points each sensor is responsible for is approximately the same for all of them, demonstrating a good distribution.

In 300 Monte Carlo experiments, for Area Coverage solutions with the aforementioned characteristics are found, we studied the objective function value normalized to the number of Anchor Points:

$$O(\mathcal{S}, \mathcal{A}) = \frac{1}{|\mathcal{A}|} \sum_{j=1}^{|\mathcal{S}|} \sum_{i=1}^{|c_j|} \frac{1}{2} \left(d_{B_{ji}}(s_j)\right)^2, \quad (27)$$

where, as before, the notation $|\mathcal{A}|$ stands for the cardinality of the set $\mathcal{A}$ and c_j are the partition of $\mathcal{A}$ according to the distances to the sensors in $\mathcal{S}$. The variation of $O(\mathcal{S}, \mathcal{A})$ with the increasing number of iterations is represented in Figure 2. The mean over the 300 trials is 75.85 with a variance of 0.58. Since the variance is low we can say that the results obtained for one arbitrary Anchor Point placement in any ROI with these sensors are a good sample of the values we might get for other placements. This value is also applicable for other ROI with the same sensor characteristics where the number of Anchor Points per sensor is approximately the same. The normalization is necessary because the number of Anchor Points varies for every found placement.

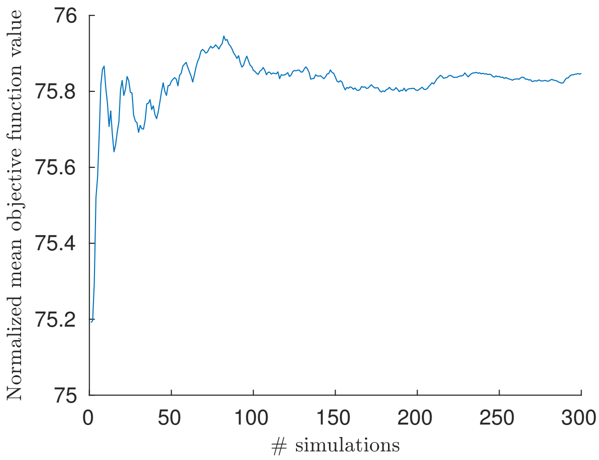

Figure 2: In 300 Monte Carlo runs the mean objective function value achieves a stability with average value 75.85 and a variance of 0.58. Therefore, representing that one arbitrary Area Coverage result is a good representative of what we might get with other Anchor Point configurations.

The repetition of the k-means++ algorithm usually leads to a better overall Anchor Point distribution and as a result a better coverage. The more times we run k-means++ the more likely it is to find a close to optimal solution of the Area Coverage mission. In the impractical case of indefinitely run it we can guarantee that it will be found.

For the cases where ε-full area coverage is found, since the Anchor Points are covered with probability τ, the objective function value is 0. It is necessary to note that having all Anchor Points covered with τ probability is not a necessary condition to achieve ε-full area coverage, however, it is sufficient.

B. Patrolling

The Patrolling mission requires the definition of a set of free parameters. The change of such parameters yields different Patrolling paths and therefore a different number of iterations until all Anchor Points are scanned at least once.

For the aforementioned parameters, 600 Monte Carlo experiments yield a mean number of simulations until all points are covered of 108 with variance 1.9×10^8. The high variance is not a good indicator, however, of the 600 simulations, only 26, which are the main contributors to this high value, took more than 150 iterations to scan every Anchor Point. In fact if the 26 simulations that took more than 150 iterations were not considered for the variance calculation its value would be only 290 which is a significant difference.

In simulations to study the differences imposed by changes in the parameters we found that as a general rule higher m, d_j and α between 0.6 and 0.7 show better overall results. The increment of m decreases the time because it makes the differences between non-covered points higher for larger Δ_i, increasing the mass w_i of such Anchor Points. The results for the range limitation d_j are already expected because smaller movement at each step requires more steps to go through an equal area, however, there is another effect to take into consideration. A high number of points to cover results in a higher region and a small d_j will sometimes create situations of balance where a sensor will move only between two positions because the masses will increase for balls that are far from that sensors position. Since it can not reach them it will not reset the masses resulting in a movement between positions where he tries to reach those high mass balls consecutively. For α, the closer to 1 the closer from the static situation and the closer to 0 the closer to the case where it is impossible to guarantee all points are covered. A compromise between the two situations is found for $\alpha \in [0.6, 0.7]$. Videos of the performed simulations can be found in `https://github.com/p-carrasqueira/k-disks`, as well as, histograms representing the number of times an Anchor Point is visited until all are visited.

Another simulation took into consideration 20 different Anchor Point placements and initial deployments where in total we placed 7064 intruders in the ROI with random positions. From all the intruders we where able to identify 68% in the first iteration, demonstrating a good overall Patrolling coverage of the ROI.

C. *Target Tracking and Escorting*

The objective of the Target Tracking and Escorting is to never lose sight of a target or intruder while he is inside the ROI. To show the algorithm performance in this mission we present Figure 3 were the path of two targets and the respective sensors performing the Escorting mission are shown.

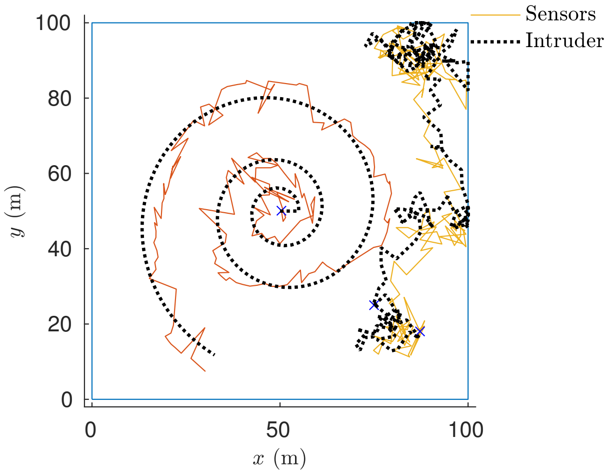

Figure 3: Target Tracking and Escorting mission with two intruders in a ROI where two agents are able to perform the Escorting mission. Videos of this example and others are found in `https://github.com/p-carrasqueira/k-disks`.

Since the definition of the path for the sensors that are not performing Tracking is equal to Patrolling it is expected that the sensors are able to patrol the ROI. In fact, in the simulations there is a constant adaptation of the network to the path of the sensor in charge of escorting an intruder.

In 300 simulations of targets with random trajectories the mission was successful in 99% of simulations with mean time until all area is covered was 118 iterations with velocity limitation of 10m/iteration, which shows that there is not a significant lost of Patrolling. The mission is successful if in 150 time steps the target is never lost.

Simulations with two targets also show that using Algorithm 2, one sensor is able to track more than one target at the same time. The network is also able to collectively adapt to escorting targets with higher velocity than each sensor.

V. Conclusions

The target management field has a wide variety of challenges that have to be overcome. Finding a solution able to solve different missions efficiently with the same formulation background is difficult. Our main achievement is the novel k-disks formulation together with the developed Anchor Point distribution technique that is able to solve three different missions using a common background.

Directions of future work are, for example: a distributed solution that allows each of the sensors to be completely autonomous, better non-symmetric Anchor Point placement, study other update coefficients models that might be better than the used or incorporating a localization algorithm for the target that allows to predict the intruders next position.

References

[1] İ. K. Altınel, N. Aras, E. Güney, and C. Ersoy. Binary integer programming formulation and heuristics for differentiated coverage in heterogeneous sensor networks. *Computer Networks*, 52(12):2419–2431, 2008.

[2] D. Arthur and S. Vassilvitskii. k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1027–1035. Society for Industrial and Applied Mathematics, 2007.

[3] A. B. Asghar and S. L. Smith. Stochastic patrolling in adversarial settings. In *American Control Conference (ACC), 2016*, pages 6435–6440. IEEE, 2016.

[4] P. Drineas, A. Frieze, R. Kannan, S. Vempala, and V. Vinay. Clustering large graphs via the singular value decomposition. *Machine learning*, 56(1-3):9–33, 2004.

[5] J.-B. Hiriart-Urruty and C. Lemarchal. *Convex analysis and minimization algorithms*. Springer-Verlag Limited, 1993.

[6] T. P. Lambrou and C. G. Panayiotou. Collaborative area monitoring using wireless sensor networks with stationary and mobile nodes. *EURASIP Journal on Advances in Signal Processing*, 2009:7, 2009.

[7] F. Li, J. Luo, W. Wang, and Y. He. Autonomous deployment for load balancing k-surface coverage in sensor networks. *IEEE Transactions on Wireless Communications*, 14(1):279–293, 2015.

[8] S. Lloyd. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2):129–137, 1982.

[9] G. Oliva, R. Setola, and M. Tesei. A stackelberg game-theoretical approach to maritime counter-piracy. *IEEE Systems Journal*, 2018.

[10] L. C. Pimenta, M. Schwager, Q. Lindsey, V. Kumar, D. Rus, R. C. Mesquita, and G. A. Pereira. Simultaneous coverage and tracking (scat) of moving targets with robot networks. In *Algorithmic foundation of robotics VIII*, pages 85–99. Springer, 2009.

[11] C. Robin and S. Lacroix. Multi-robot target detection and tracking: taxonomy and survey. *Autonomous Robots*, 40(4):729–760, 2016.

[12] L. Vandenberghe. Fast proximal gradient methods (fista). Lecture Notes. Visited July 2018.

[13] A. E. Xavier and A. A. F. De Oliveira. Optimal covering of plane domains by circles via hyperbolic smoothing. *Journal of Global Optimization*, 31(3):493–504, 2005.

[14] Q. Yang, S. He, J. Li, J. Chen, and Y. Sun. Energy-efficient probabilistic area coverage in wireless sensor networks. *IEEE Transactions on Vehicular Technology*, 64(1):367–377, 2015.

[15] Y. Zou and K. Chakrabarty. A distributed coverage- and connectivity-centric technique for selecting active nodes in wireless sensor networks. *IEEE Transactions on Computers*, 54(8):978–991, 2005.