Constrained Coverage Using Switching Formation - Extended Abstract

Yoav Palti* and Dan Zelazo†

I Background

The mission of monitoring some area using sensors is widely spread in various fields - it can be used for photographing a large area, receiving or sending electromagnetic signals, tracking targets, or even for designing an algorithm for a robotic vacuum cleaner [1–3]. This research area is known in the scientific community as the *coverage control* problem [4]. Sensor coverage can be generally described as the reflection of how well a given range (for example, area) is monitored by sensors. This field is well explored in the scientific community, especially in the recent years when the fields of distributed algorithms and cooperative systems were developed.

When coverage is required for large areas, and specifically when a single sensor cannot provide full coverage of that area, the idea of using a *formation* of mobile sensors is presented [5]. The main advantage of using a formation is the ability to perform a task in a distributed way, which leads to cheaper sensors, objective flexibility, and survivability (the system can function even if one of the sensors is broken) [6].

A concept that repeats in many research studies is finding a set of trajectories (at least one route for every sensor), allowing the mobile sensors to achieve the required coverage goal. For example, when trying to cover an area using two mobile sensors, the algorithm will define a trajectory for each sensor such that the whole area will be covered [7, 8]. Another idea that appears in most of the relevant literature is the use of Voronoi diagrams for optimizing the sensor location [6, 8].

In this research, we developed a new coverage algorithm. Using this algorithm, the operator can switch formations on different parts of the area and get a static coverage on some part of the whole area. Moreover, we defined a coverage constraint, that defines an area that at least one agent has to be present in at all steady-state times. This constraint represent, for example, an agent that must be within the home-station coverage radius.

I.A Voronoi Diagrams

The basic mathematical concept that makes out solution possible is the *Voronoi Diagram*. While being a method to partition an area with some cost function, the is a widely-used representation in the coverage problem [6, 8, 9]. The Voronoi Diagram of a region $\Omega \subset \mathbb{R}^2$ is the set of partitions $\mathcal{V} = \left\{V_i \mid \cup V_i = \Omega\right\}$, generated by the generators $\mathcal{Z} = \{z_1, \ldots, z_n \mid z_i \in \Omega\}$, such that:

$$V_i = \{q \in \Omega \mid \|q - z_i\| \leq \left\|q - z_j\right\| \forall z_i, z_j \in \mathcal{Z}\},$$

Where V_i corresponds to the i-th element of $\mathcal{Z}$, and $\|\cdot\|$ denotes the Euclidean distance.

*Faculty of Aerospace Engineering, Technion - Israel Institute of Technology, Haifa, yoavp10@gmail.com

†Associate Professor, Faculty of Aerospace Engineering, Technion - Israel Institute of Technology, Haifa, dzelazo@technion.ac.il

Manuscript 60 submitted to The 59th Israel Annual Conference on Aerospace Sciences. Received October 6, 2018.

One can define a density function each Voronoi partition V_i, ρ_i. Then, we can define the center of mass for each partition:

$$z_i^* = \frac{\int_{V_i} y\rho(y)dy}{\int_{V_i} \rho(y)dy}. \quad (1)$$

If a generator $z_i = z_i^* \forall V_i$, we call this partitioning a *centroidal Voronoi tessellation* (CVT). This such of tessellations are useful in terms of location optimization [6,7,9]. A Voronoi tessellation illustration can be seen in figure 1.

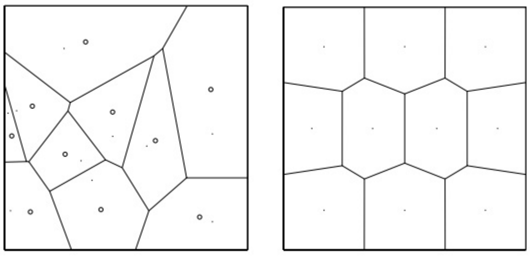

Figure 1: An illustration of Voronoi diagrams. On the left, a regular Voronoi diagram, and on the right - a centroidal Voronoi tessellation. The figure is taken from [9].

I.B Lloyd's Algorithm

As mentioned above, the CVT's are very useful for locational optimization. However, calculating the CVT might be a complicated task. Lloyd proposed a very simple way of calculating the CVT [10]:

1. Calculate the Voronoi diagram for the current agents positions.
2. Calculate the center of mass for every cell
3. move the agents the center of mass
4. repeat until converge

Cortes et al. proposed a control algorithm based on Lloyd's algorithm [6]. According to [6], if we define agent i position as p_i and the i's partition centroid as C_{V_i}, then for some proportional constant k_{prop}, the controller can be defined as:

$$u_i = -k_{prop}\left(p_i - C_{V_i}\right)$$

Another thing to notice is that one can modify the density function, which will change the geometrical position of the centroid.

II Problem Definition

Let us consider an area $A \subset \mathbb{R}^2$ that we aim to cover with a number of mobile sensors. The solution will be based on configurations (which are like the concept of frameworks). Given a set of sensors $S = \{s_1 \dots s_n\}$ located in positions $p_i \in \mathbb{R}^2$ (for $i = 1, \dots, n$) at time t, a configuration $c(t)$ is stacking the sensor positions,

$$c(t) = \begin{bmatrix} p_1^T(t) & \cdots & p_n^T(t) \end{bmatrix}^T \in \mathbb{R}^{2n}. \quad (2)$$

Manuscript 60 submitted to The 59th Israel Annual Conference
on Aerospace Sciences. Received October 6, 2018.

Each sensor i has a coverage radius R (assuming all sensors are identical), so it can cover a disk $D\left(p_i\left(t\right), R\right) \subset \mathbb{R}^2$, centered at $p_i\left(t\right)$,

$$D\left(p_i\left(t\right), R\right) = D_i\left(t\right) = \{x \in \mathbb{R}^2 \mid \left\|x - p_i\left(t\right)\right\|_2 \leq R\}. \tag{3}$$

The set of all possible configurations is defined as $\mathtt{C_a} = \left\{c_i\left(t\right)\right\} \in \mathbb{R}^{2r}$. Where $c_i(t)$ is the i-th configuration of the configurations set, and the total area that the i-th configuration can cover is

$$A_{cov}\left(c_i(t)\right) = \cup_i D_i\left(t\right) \subset \mathbb{R}^2. \tag{4}$$

We also require that the configurations satisfy a *Coverage Constraint*: A given area inside the area $A_m \subset A$ must be covered always (e.g. ground station). This condition, named *Area Constraint Condition*, defines a set of possible configurations $\mathtt{C_{acc}} = \left\{c_i\left(t\right) \mid A_m \subset A_{cov}\left(c_i\left(t\right)\right)\right\}$.

III Proposed Solution

Assuming that there are no constraints at all, we propose an algorithm to solve the problem mentioned above:

1. Using some random initial guess, calculate the CVT for the whole area
2. for each partition (assuming that the agents can actually cover each partition with their coverage radius), calculate the CVT. The initial positions for the CVT calculation is the previous partition CVT

We should notice the following things:

1. In stage 1, the number of the partition that the area will be divided to is somewhat arbitrary.
2. In stage 2, we assume that the coverage radius of the agents is big enough to cover the whole partition. If it's not the case, then we can dived the area for more partitions in stage 1. Some more advanced work can propose an algorithmic solution for this problem.
3. The Voronoi diagram can be calculated even if the agents are outside of the required area. It does require sometimes more iterations.

However, as one can see, we didn't answer the area constraint.

III.A Projected Lloyd's Algorithm

An approach to ensure the area constraint is to tweak the original Lloyd's algorithm as such:

1. Calculate the Voronoi diagram for the current agents positions.
2. Calculate the center of mass for every cell
3. Project the center of mass of every cell to the area constraint limiting polygon
4. move the agents the projected center of mass
5. repeat until converge

As one can see, we added one more step - where we project the cell center of mass to the polygon that defines the area constraint. This step allows us to ensure that at least one agent will be *on* the limiting polygon, thus we will have coverage within the constraint.

III.A.1 and Stability of the Projected Lloyd's Algorithm

in [6], we can find a proof for the stability of Lloyd's algorithm in the form of controller. We added one linear step to this solution, and it is very easy to see that this does not affect the Lyaponov function, therefore the system is asymptotically stable.

III.B Results

First of all, we shall show the difference between the constrained and non constrained CVT:

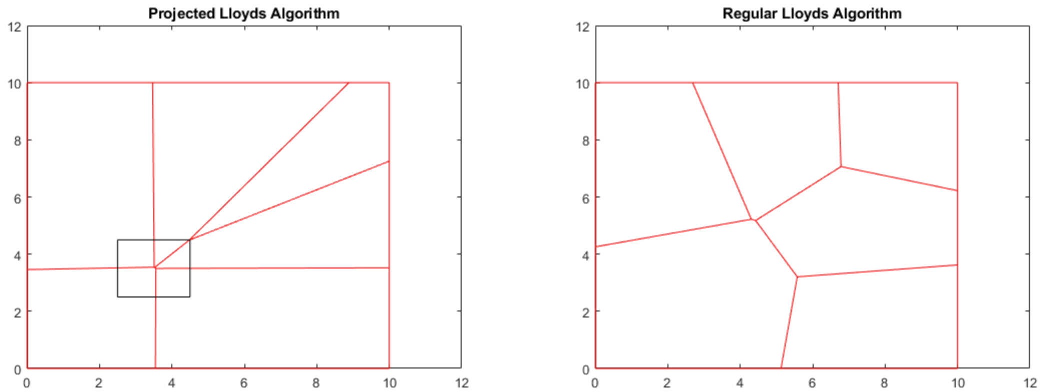

Figure 2: Projected vs. Regular Lloyd's algorithm

Next thing is to see the different agents formations within each area partition. The following example is for the projected case, and the order of the partitions that are being covered is arbitrary (i.e. in step $i + 1$ the partition covered is not necessarily adjacent to the once that was covered in step i): First of all, we shall show the difference between the constrained and non constrained CVT:

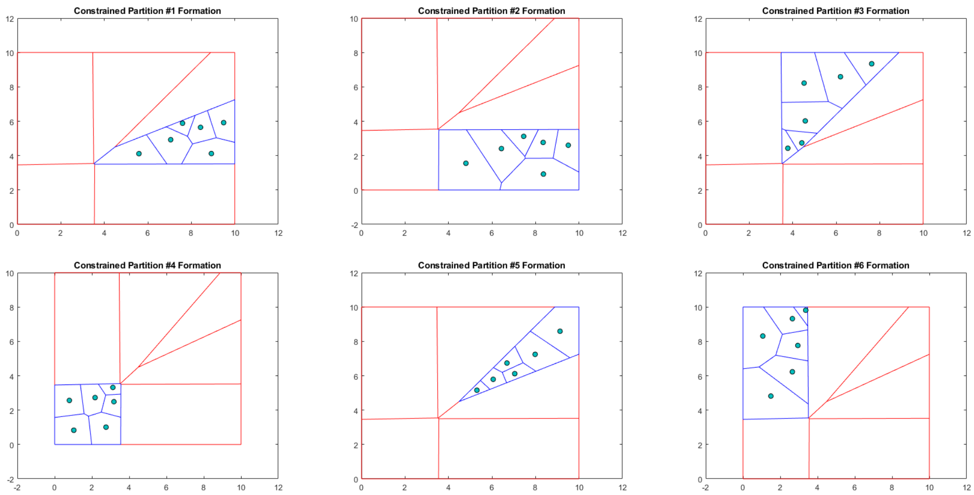

Figure 3: Covering different partitions in an arbitrary order

References

[1] Nigam, N., Bieniawski, S., Kroo, I., and Vian, J., "Control of multiple UAVs for persistent surveillance: Algorithm and flight test results," *IEEE Transactions on Control Systems*

Technology, Vol. 20, No. 5, 2012, pp. 1236–1251, doi:10.1109/TCST.2011.2167331.

[2] Palacios-Gasós, J. M., Montijano, E., Sagüés, C., and Llorente, S., "Multi-robot persistent coverage with optimal times," in "2016 IEEE 55th Conference on Decision and Control (CDC)," , 2016, pp. 3511–3517, doi:10.1109/CDC.2016.7798796.

[3] Loizou, S. G. and Constantinou, C. C., "Multi-robot coverage on dendritic topologies under communication constraints," in "2016 IEEE 55th Conference on Decision and Control (CDC)," , 2016, pp. 43–48, doi:10.1109/CDC.2016.7798244.

[4] Cassandras, C. G. and Li, W., "Sensor Networks and Cooperative Control," *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC '05. 44th IEEE Conference on*, pp. 4237–4238.

[5] Hokayem, P. F., Stipanović, D., and Spong, M. W., "On persistent coverage control," *Proceedings of the IEEE Conference on Decision and Control*, pp. 6130–6135, doi:10.1109/CDC.2007.4434875.

[6] Cortes, J. and Martinez, S., "Coverage control for mobile sensing networks," *Robotics and Automation, ...*, Vol. 20, No. 2, 2004, p. 13, doi:10.1109/TRA.2004.824698.

[7] Atinç, G. M., Stipanović, D. M., Voulgaris, P. G., and Karkoub, M., "Supervised coverage control with guaranteed collision avoidance and proximity maintenance," *Proceedings of the IEEE Conference on Decision and Control*, pp. 3463–3468, doi:10.1109/CDC.2013.6760414.

[8] Hussein, I. I. and Stipanovic, D. M., "Effective Coverage Control for Mobile Sensor Networks With Guaranteed Collision Avoidance," *IEEE Transactions on Control Systems Technology*, Vol. 15, No. 4, 2007, pp. 642–657, doi:10.1109/TCST.2007.899155.

[9] Du, Q., Faber, V., and Gunzburger, M., "Centroidal Voronoi Tessellations: Applications and Algorithms," *SIAM Review*, Vol. 41, No. 4, 1999, pp. 637–676, doi:10.1137/S0036144599352836.

[10] Lloyd, S., "Least squares quantization in PCM," *IEEE Transactions on Information Theory, Information Theory, IEEE Transactions on, IEEE Trans. Inform. Theory*, Vol. 28, No. 2, 1982, pp. 129–137, doi:10.1109/TIT.1982.1056489.