A Projected Lloyd's Algorithm for Coverage Control Problems

Yoav Palti* and Daniel Zelazo†

This work presents a new approach for a coverage problem for large areas and a limited number of sensors. We consider the problem where there are not enough sensors to provide complete coverage of the designated area at once, and therefore the sensors should coordinate their coverage in a sequential manner. Furthermore, we require that at each stage of coverage, a sub-region in the area is constantly monitored. Our solution approach is a combination of two algorithms, the first one is Lloyd's algorithm, which is well known and used in this field, and the second one is a modified Lloyd's algorithm that we term the *projected Lloyd's Algorithm*. We demonstrate our results with numerical simulations.

I Introduction

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 [**?**, **?**, **?**]. This research area is known in the scientific community as the *coverage control* problem [**?**]. 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 [**?**]. 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) [**?**].

A concept that repeats in many researches 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 [**?**, **?**]. Another idea that appears in most of the relevant literature is the use of Voronoi diagrams for optimizing the sensor location [**?**, **?**].

To the best of out knowledge, there isn't any solution that tries to deal with the coverage problem in the scenario where, on the one hand, an operator has to maintain coverage of a specific part of the area, and on the other hand, have a set of formations that all together provides a complete coverage of the area. In this research, we developed a new coverage algorithm that deals with this problem. The operator can switch formations on different parts of the area and get a static coverage on some part of the whole area. For this algorithm, we defined a coverage constraint, that defines an area that at least one agent has to be present in at all steady-state

*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

times. This constraint represent, for example, an agent that must be within the home-station coverage radius.

Following this chapter, this extended abstract will include a brief background about Voronoi Diagrams and general coverage algorithms. After it, the problem will be formally formulated, and later a solution will be introduced.

II Voronoi Diagrams and Distributed Coverage Algorithms

In this chapter we will introduce two main concepts that were used in this work. First, Voronoi Diagrams, and later Lloyd's algorithm.

II.A Voronoi Diagrams

The basic mathematical concept that we employ in this work 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 [**?**, **?**, **?**]. The Voronoi Diagram of a region $\Omega \subset \mathbb{R}^2$ is the set of partitions $\mathcal{V} = \{V_i \mid \cup V_i = \Omega\}$, generated by the generators $\mathcal{Z} = \{z_1, \ldots, z_n \mid z_i \in \Omega\}$, such that

$$V_i = \{q \in \Omega \mid \|q - z_i\| \leq \|q - z_j\| \, \forall z_i, z_j \in \mathcal{Z}\}, \tag{1}$$

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

A more intuitive and non-formal definition of the Voronoi Diagram is as follows. If we take some area and place points p_i in it, then each partition is the set of all points that are closer to p_i than p_j, when $j \neq i$. An example is the problem where we have post offices in some city, and we need to decide which house will be served by which post office branch. Using Voronoi partitioning, we can determine which houses are the nearest to each branch and therefore conclude which houses each branch will serve.

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

$$z_i^* = \frac{\int_{V_i} y\rho(y)dy}{\int_{V_i} \rho(y)dy}. \tag{2}$$

If a generator $z_i = z_i^* \, \forall \, V_i$, we call this partitioning a *centroidal Voronoi tessellation* (CVT). Such tessellations are useful in terms of location optimization [**?**, **?**, **?**]. A Voronoi tessellation illustration can be seen in Figure 1. Going back to the post-office example, the CVT can help us determine where those post office branches should be located for an optimal distribution of houses per branch.

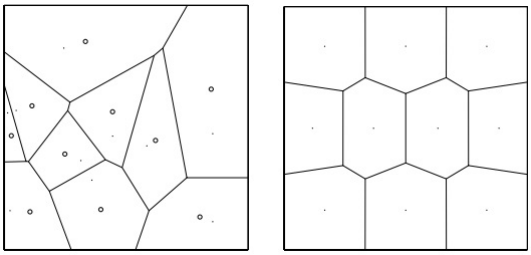

Figure 1: An illustration of Voronoi diagrams. On the left, a regular Voronoi diagram, and on the right, a centroidal Voronoi tessellation. The figure is taken from [**?**].

As can be seen in equation (1), the calculation of each Voronoi cell for some agent depends on the other agents positions. Therefore, any sort of calculation required *sharing information*. This can be done in a centralized approach (as it was done for this research), but there exists also decentralized calculations, for example [**?**]. Using decentralized calculation, each agent have to have the info only on their neighbours (or any limited amount of agents), which allows for calculating huge networks, in dynamic situations where agents can, for example, connect and disconnect from the network in real time.

II.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 [**?**], presented in Alogirthm 1.

Algorithm 1 Lloyd's Algorithm

1: Calculate the Voronoi diagram for the current agents positions.
2: Calculate the center of mass for every cell.
3: Move the agents to the center of mass.

Cortes et al. proposed a control algorithm based on Lloyd's algorithm [**?**]. According to [**?**], 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:

$$u_i = -k_{prop}\left(p_i - C_{V_i}\right) \tag{3}$$

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

III Problem Formulation

Let us consider an area $A \subset \mathbb{R}^2$ that we aim to cover with $n \in \mathbb{N}$ *mobile* sensors. Those sensors are defined as the set $S = \{s_1, \ldots, s_n\}$ located at positions $P(t) = \{p_i(t) \in \mathbb{R}^2 \mid i = 1, \ldots, n\}$:

- The mobile sensors are modelled using integrator dynamics $\dot{p_i}(t) = u$.
- Each sensor can cover an area described by the abstract set $C_i\left(p_i(t)\right) \subset \mathbb{R}^2$.

III.A Configuration

A configuration c of the mobile sensors at time t is the stack of the sensor positions at time t:

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

Thus the coverage of a configuration:

$$D\left(c\left(t\right)\right) = \cap C_i(p_i(t))$$

Let us consider a sub-area of A, $A_m \subset A$ which must be partially covered (e.g. ground station).

Assumption 1. $D\left(c\left(t\right)\right) \subset A$ - *a single configuration* can't *provide full coverage!*

III.B Partition

The partition of $A \in \mathbb{R}^2$, $PR(A)$, is a finite set built from n subset $pr_i \subset \mathbb{R}^2,\ i = 1,\ \ldots,\ n$ such that:

- $pr_i \cap pr_j = \emptyset\ \forall i \neq j$.
- $\cup pr_i = A$.

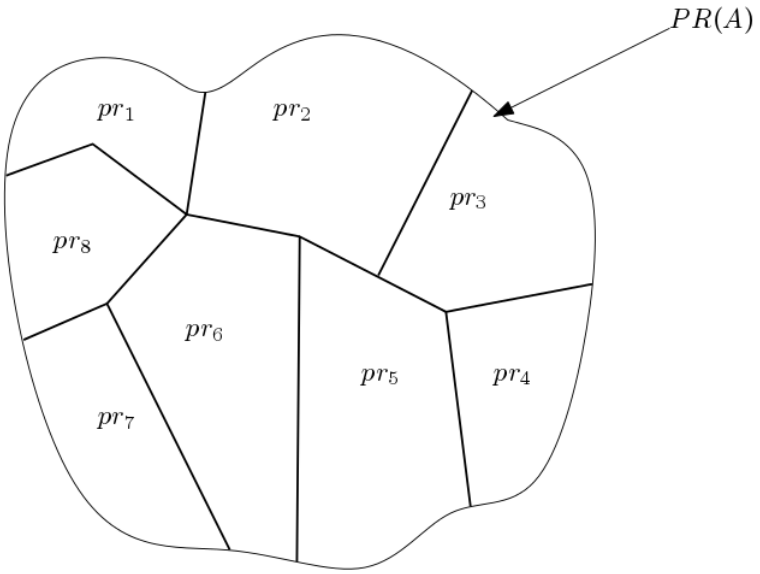

Figure 2: An example of a partition of some area $PR(A)$

III.C Problem Definition

Problem 1. *1. Find a partition $PR(A)$ such that each subset $pr_i \in PR(A),\ pr_i \cap A_m \neq \emptyset$*

2. Find a deployment controller, such that

$$pr_i \subseteq \lim_{t \to \infty} D(c(t)), \textit{for } i = 1, \ldots, n.$$

IV Proposed Solution

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

Algorithm 2 Problem (1) Solution Algorithm

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.

IV.A Projected Lloyd's Algorithm

An approach to ensure the area constraint is to modify the original Lloyd's algorithm. If we use the *Projection Operator* on the cell's center of mass, then the control law 3 will be now:

$$u_i = -k_{prop}\left(p_i - \textit{proj}\left(C_{V_i}\right)\right) \tag{4}$$

Formulating this as an algorithm:

Algorithm 3 Projected Lloyd's Algorithm

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 two more steps over the original Lloyd's algorithm - where we project the cell center of mass to the polygon that defines the area constraint (steps number 3-4), and then moving the agents to those points. Those step allows us to ensure that at least one agent will be *on* the limiting polygon, thus we will have coverage within the constraint.

IV.A.1 Stability of the Projected Lloyd's Algorithm

Theorem 1. *The controller proposed in (4) is locally asymptotical stable.*

Full proof will be given in the final version. However, in [**?**], 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.

IV.B Results

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

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:

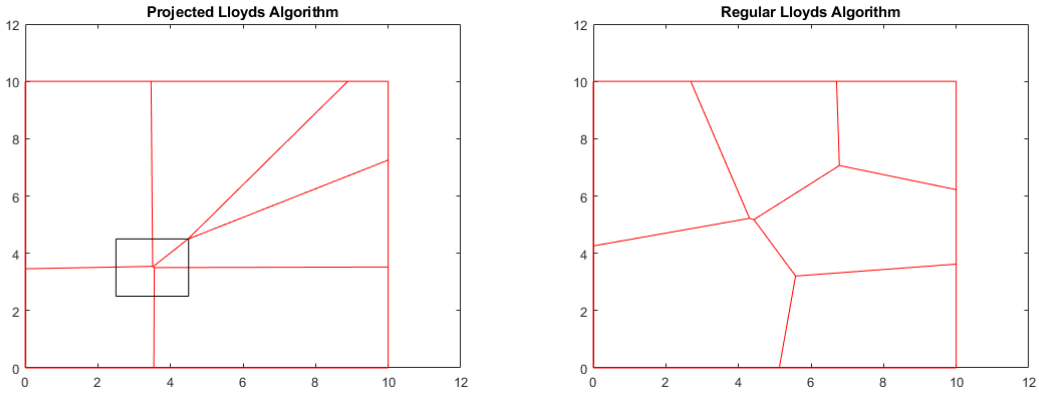

Figure 3: Projected vs. Regular Lloyd's algorithm

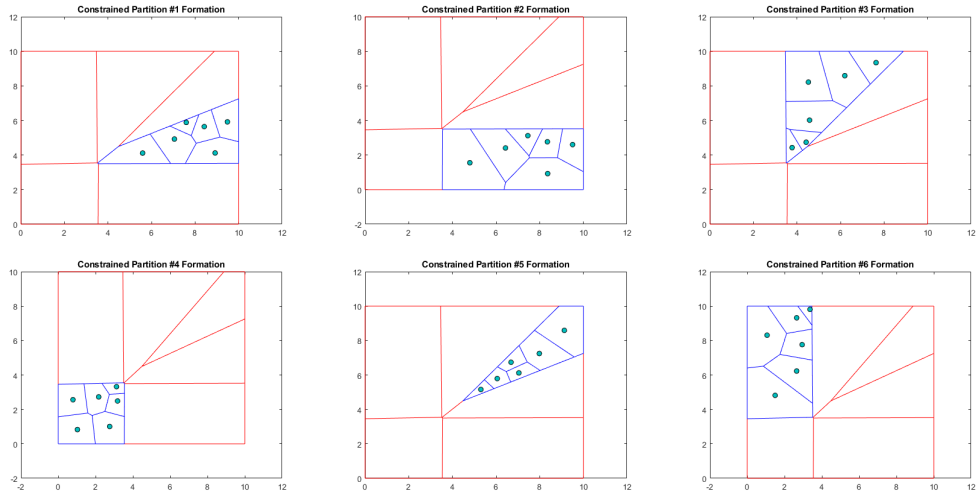

Figure 4: Covering different partitions in an arbitrary order

V Concluding Remarks

The Coverage Problem is well known and researched. This work provides a new approach for the common problem of not having enough sensors, and deals with it from a static point of view, while maintaining coverage on some constrained area.
The proposed solution is a simple control law which, under the work assumption, solves the problem. The simplicity of the control law, all together with the fact its stability, open the way for combining it with other control laws in some future work, which might lead to some interesting results. For example, combining this controller with a formation control controller, together with calculating the Voronoi in a distributed way, can lead to some interesting results for autonomous, dynamically changed swarms.

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] Adams, J. S., Sun, W., and Chen, Y. Q., *Formations with decentralized Centroidal Voronoi tessellation algorithm*, Vol. 42, IFAC, 2009, doi:10.3182/20091006-3-US-4006.0021.

[11] 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.