Project for IOT Class A.Y. 2022/2023

{bartolini, prata, trombetti}@di.uniroma1.it

A Patroling Protocol for UAV Networks

For this project, we will focus on the cooperation of UAVs (Unmanned Aerial Vehicles) in a 3D environment. The task is to plan trajectories of the UAVs so that cameras onboard the UAVs inspect a set of N unique inspection points. All UAVs have a predefined starting position. The objective of the task is to minimize the time of inspection while capturing all the inspection points. An already working solution is provided as a part of the assignment. However, this example solution has poor performance and can be improved significantly.

Task Overview You are given a set of UAVs $\mathcal{U}$ and that are required to inspect a set of inspection points (IPs) in the set $\mathcal{I}$ to optimize their age of information and keep it below their desired inter-visit threshold. We define *age of information (AoI)* at time t, the function $a_p(t)$ representing the time (seconds) elapsing between two consecutive visits of any UAV to an IP point $p \in \mathcal{I}$. We define the *inter-visit threshold* as the function $\theta(p)$ representing the AoI (seconds) that the IP p can tolerate. At any point in time t, if $a_p(t) > \theta(p)$ a *violation* of the IP just happened.

A *visit* or *inspection* to an IP p of a UAV u is considered *valid* if an only if $d(\dot{p}, \dot{u}) \le r_s$ where $d(\cdot, \cdot)$ is the euclidean distance between $\dot{p}, \dot{u}$ coordinates in the euclidean space of the IP and the UAV, r_s the sensing range of a UAV. In other words, a visit is considered successful if on only if the UAV lands on the IP at enough distance.

Your task is to design a patrolling policy $\pi_u : \mathcal{R} \to \mathcal{I} \quad \forall\, u \in \mathcal{U}$ that is a function mapping a time step to an inspection point, that is the next inspection point to visit.

Optimizing Patrolling of an IP A patrolling instance is a 6-tuple $(\mathcal{U}, \mathcal{I}, a, \theta, f, \pi)$ and can be optimized under a series of criteria. The AoI of an IP can be defined in time as the sawtooth wave $a_p(t)$ in Figure 1, the function values 0 as soon as a valid visit to the IP is done. We will assume no hovering time for the inspection. We will go though a series of optimization criteria $c_i(p)\ \forall\, p \in \mathcal{I}$ that a patrolling policy may want to *minimize*.

It follows a list of criteria:

- A patrolling policy can be such to ***minimize cumulative AoI*** of the IP so to keep the sensed information as fresh as possible. This objective corresponds in minimizing the yellow area in Figure 1. Formally corresponds to the following:

$$c_1(p) : \int_{t=0}^{M} a_p(t)\, dt \tag{1}$$

  where M is the mission duration expressed in seconds.

- When adopting inter-visit threshold θ the objective is to ***minimize cumulative time in violation***, that corresponds in minimizing the red area in Figure 2. In absence of violations, it additionally consists in minimizing the cumulative age of information as a secondary task (weighted by ε), formally:

$$c_2(p) : \int_{t=0}^{M} a_p(t) \cdot \rho(a_p(t) > \theta_p)\, dt + \varepsilon \cdot \int_{t=0}^{M} a_p(t)\, dt \tag{2}$$

  where $\rho(\cdot)$ returns 1 if the input predicate is true, 0 otherwise. Thus $\rho(a_p(t) > \theta_p)$ is 1 if the AoI of the IP is above its inter-visit threshold, 0 otherwise.

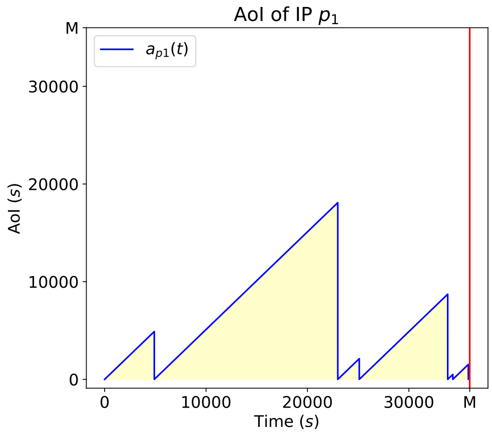


Figure 1: AoI without constraint

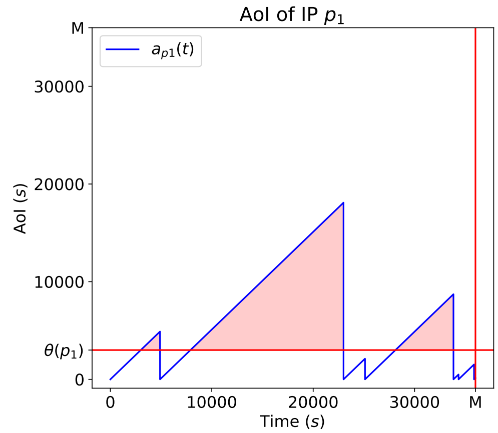


Figure 2: AoI with constraint

- Another optimization could be to ***minimize maximum AoI***, formally:

$$c_3(p) : \max_{0 \leq t \leq M} a_p(t) \tag{3}$$

- Finally one could ***minimize number of violations***, formally:

$$c_4(p) : \int_{t=0}^{M} \rho(\rho(a_p(t-1) \leq \theta_p) \wedge \rho(a_p(t) > \theta_p))\, dt \tag{4}$$

Optimizing Patrolling of Multiple IPs Given a policy π, it can handle the IP visits in several different ways, causing, or caring about unfairness among the them. Optimizing for any of the two approaches deeply depends on the scenario in consideration. How should it handle the IPs distributions? Minimize the:

- **average** criterion over the IPs:

$$\min \frac{1}{|\mathcal{I}|} \sum_{p \in \mathcal{I}} c_i(p) \tag{5}$$

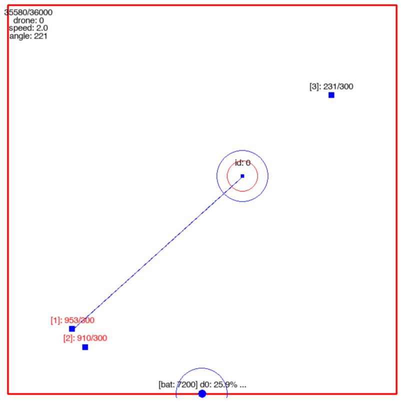
- **maximum** (worst) criterion over the IPs:

$$\min \max_{p \in \mathcal{I}} c_i(p) \tag{6}$$

Let us make an example, to understand the impact of one of the above metrics on the fairness of the policy. Let us define the scenario $M = 5400\, s$ (1h 30m), $|\mathcal{U}| = 1$, $|\mathcal{I}| = 3$, $\theta(p) = 300\, s\ \forall p \in \mathcal{I}$, with nodes displaced as in Figure 3a. Let policy π_{loop} visiting IPs 1 and 2 in a loop, neglecting IP 3 completely; and policy π_{tsp} visiting all three IPs with a Travelling Salesman Problem (TSP) tour.

Figures 3b and 3c show the score board of these two baseline policies. Notice in Figure 3b how π_{loop} does very good at minimizing average c_2 (integral of the delay or total delay) over the IPs, compared to π_{tsp} in Figure 3c: 1500 s versus 4014 s of average total delay. This happens because looping over IPs 1 and 2 lowers the average significantly as they are very close to each other. The average hides the impact of the important violation on IP 3, hiding the unfairness of the policy.

Notice that for π_{loop} when M is large, the contribute of IP 3 will be so high to increase the average significantly and make it less preferable.



(a) IPs displacement in basic scenario.

	C1: INTEGRAL AOI	C2: INTEGRAL DELAY	C3: MAX AOI	C4: NUMBER VIOLATIONS	C?: MAX DELAY
0	165099.0	0	281.0	0	0.0
1	158874.0	0	256.0	0	0.0
2	14577300.0	5100	5399.0	1	5399.0

	C1: INTEGRAL AOI	C2: INTEGRAL DELAY	C3: MAX AOI	C4: NUMBER VIOLATIONS	C?: MAX DELAY
count	3.000000e+00	3.000000	3.000000	3.000000	3.000000
mean	4.967091e+06	1700.000000	1978.666667	0.333333	1799.666667
std	8.322686e+06	2944.486373	2962.121931	0.577350	3117.114103
min	1.588740e+05	0.000000	256.000000	0.000000	0.000000
25%	1.619865e+05	0.000000	268.500000	0.000000	0.000000
50%	1.650990e+05	0.000000	281.000000	0.000000	0.000000
75%	7.371200e+06	2550.000000	2840.000000	0.500000	2699.500000
max	1.457730e+07	5100.000000	5399.000000	1.000000	5399.000000

(b) π_{loop} score board

	C1: INTEGRAL AOI	C2: INTEGRAL DELAY	C3: MAX AOI	C4: NUMBER VIOLATIONS	C?: MAX DELAY
0	3698245.0	4200	1583.0	4	1583.0
1	3363231.0	3943	1352.0	4	1352.0
2	3248781.0	3900	1352.0	5	1352.0

	C1: INTEGRAL AOI	C2: INTEGRAL DELAY	C3: MAX AOI	C4: NUMBER VIOLATIONS	C?: MAX DELAY
count	3.000000e+00	3.000000	3.000000	3.000000	3.000000
mean	3.436752e+06	4014.333333	1429.000000	4.333333	1429.000000
std	2.335776e+05	162.223097	133.367912	0.577350	133.367912
min	3.248781e+06	3900.000000	1352.000000	4.000000	1352.000000
25%	3.306006e+06	3921.500000	1352.000000	4.000000	1352.000000
50%	3.363231e+06	3943.000000	1352.000000	4.000000	1352.000000
75%	3.530738e+06	4071.500000	1467.500000	4.500000	1467.500000
max	3.698245e+06	4200.000000	1583.000000	5.000000	1583.000000

(c) π_{tsp} score board

The score boards also highlight that the minimization of the maximum c_2 criterion over the IPs (Equation 6) better highlights the unfairness of π_{loop} policy, scoring $5100\,s$ (i.e., $M - \theta(3)$, IP 3 was never visited) against $4200\,s$ of π_{tsp}.

Measures of Fairness of a Policy Other approaches consist in minimizing:

- **standard deviation** criterion over the IPs:

$$\min \operatorname*{std}_{p \in \mathcal{I}} c_i(p) \tag{7}$$

- **min/max** criterion over the IPs:

$$\min \max_{p \in \mathcal{I}} c_i(p) - \min_{p \in \mathcal{I}} c_i(p) \tag{8}$$

notice that these metrics alone would consider preferable situations where the AoI levels are very close (even if they are very high). This because they focus more on the spread of the distribution rather than the delays themselves.

Assignment We want your solution to focus on the optimization of a series patrolling scenarios.

- **scenario 1**: $\mathcal{U}$, $\mathcal{I}$, $\theta(p) = M\ \forall p \in \mathcal{I}$, that is all the UAVs can visit all the IPs and they have identical inter-visit threshold.
- **scenario 2**: $\mathcal{U}$, $\mathcal{I}$, $\theta(p) = rand\ \forall p \in \mathcal{I}$, that is all the UAVs can visit all the IPs and they have different inter-visit threshold.

For each of the subtasks you will be given 3 *validation* scenarios for validating your proposed solution. Each scenario varies in the **number of UAVs**, **IPs positions** and **number**. We test your solution on several *test* scenarios and sum the score you obtain in each of the scenarios and in each task.

To score, you are supposed to propose a solution that performs well both under a **freshness** and **fairness** point of view. So we will provide you with two scores, one is fond on $c_2(p)$ as:

$$\frac{1}{|\mathcal{I}|} \sum_{p \in \mathcal{I}} \int_{t=0}^{M} a_p(t) \cdot \rho(a_p(t) > \theta_p)\, dt + \varepsilon \cdot \int_{t=0}^{M} a_p(t)\, dt \tag{9}$$

the fairness score will be:

$$\max_{p \in \mathcal{I}} c_2(p) - \min_{p \in \mathcal{I}} c_2(p) \tag{10}$$

The winner of the competition will be the group with the highest score.

References Clustring Algorithms[1]. Travelling Salesperson Problems [2].

[1] https://scikit-learn.org/stable/modules/clustering.html
[2] https://towardsdatascience.com/solving-travelling-salesperson-problems-with-python-5de7e883d847