Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Computer Science 19 (2013) 80 – 87

Procedia Computer Science

The 4th International Conference on Ambient Systems, Networks and Technologies (ANT 2013)

Energy efficient mobile target tracking using flying drones

Dimitrios Zorbas[a], Tahiry Razafindralambo[a,b], Di Puglia Pugliese Luigi[c], Francesca Guerriero[c]

[a]Inria Lille - Nord Europe – France
[b]Inria Chile – Chile
[c]Department of Mechanical, Energy and Management Engineering – University of Calabria – Italy

Abstract

This paper focuses on the energy efficiency problem where camera equipped flying drones are able to detect and follow mobile events that happen on the ground. We give a mathematical formulation of the problem of minimizing the total energy consumption of a fleet of drones when coverage of all events is required. Due to the extremely high complexity of the binary optimization problem, the optimum solution cannot be obtained even for small instances. On the contrary, we present LAS, a localized solution for the aforementioned problem which takes into account the ability of the drones to fly at lower altitudes in order to conserve energy. We simulate LAS and we compare its performance to a centralized algorithm and to an approach that uses static drones to cover all the terrain. Our findings show that LAS performs similar to the centralized algorithm, while it outperforms the static approach by up to 150% in terms of consumed energy. Finally, the simulation results show that LAS is very sustainable in presence of communication errors.

Selection and peer-review under responsibility of Elhadi M. Shakshuki

Keywords: Target tracking, Unmanned aerial vehicle, Drone, Energy efficiency

1. Introduction

During the last years, an increased use of flying drones has been noticed. The invention of light materials, low energy consumption machines and high performance processing units led to the construction of flexible flying robots. They can be used in a variety of applications such as the vehicle tracking, the traffic management and the fire detection [1]. The drones are able to fly autonomously in different altitudes and they are usually equipped with sensors to monitor the environment and communication units to exchange data with other drones or central stations.

Target tracking and vehicle control are two well studied problems in the literature. However, most of the previous works focus on the quality of detection of the mobile targets from a single [2, 3, 4] or multiple vehicles [5, 6] and there is no mention about energy efficiency. In more detail, the authors propose techniques to construct a control strategy through which they maintain good visibility of the targets on the

Email addresses: dimitrios.zormpas@inria.fr (Dimitrios Zorbas), tahiry.razafindralambo@inria.fr (Tahiry Razafindralambo), ldipuglia@deis.unical.it (Di Puglia Pugliese Luigi), guerrier@unical.it (Francesca Guerriero)

1877-0509
Selection and peer-review under responsibility of Elhadi M. Shakshuki
doi:10.1016/j.procs.2013.06.016

ground. All the fundamental tools of how a target can be detected, how to takeoff and how to land a drone are well analyzed in the literature [7, 8, 9].

In this paper, we focus on providing a power efficient and reliable drone scheduling by adjusting the drones position ensuring the surveillance of all the targets at the same time. To the best of our knowledge, this is the first paper that deals with energy efficiency and mobile target tracking. We assume drones that are equipped with a camera to detect mobile events. This camera has a maximum vision angle and a maximum range beyond which the robot cannot accurately detect the events. The drones fly higher in order to monitor a larger area and detect more events. However, the higher a drone flies, the more energy it consumes.

To minimize the energy consumption we allow the drones to adjust their altitude. Intuition says that putting all the drones in minimum altitude we can obtain the best solution. However, when multiple events are close enough to each other, this solution is not efficient. A formal description of the problem is given, however its solution requires an extremely huge amount of time even for tiny instances. For that reason, an efficient localized approach is presented along with a centralized heuristic for comparison purposes.

The rest of the paper is organized as follows. In Section 2, we provide a detailed description of the problem. Our target tracking solutions are presented in Section 3, while in Section 4 we evaluate our methods and we present extensive simulation results. Finally, Section 5 concludes the paper.

2. Problem description

The system is modeled as a 3-dimensional terrain with axes X, Y, and Z. A set of targets $T = \{t_1, t_2, \cdots, t_n\}$ can be randomly moved with speed υ_i, $\forall\, i \in [1, n]$ using X and Y axes.

Drones are used to cover the targets in T. Each drone is equipped with a camera aiming at the ground. The camera has a maximum angle of view θ. We assume that the field of view is a cone with height h and angle θ. The flat surface of the cone represents the monitoring area and every target that lies in this area can be covered by the drone. The latter can move higher or lower in order to increase or decrease the size of the monitoring area.

The altitude of the drones can be controlled by adjusting the power level of the motors. According to this observation we assume that the higher the distance h, the higher the energy consumption. The energy consumption of a drone u at any time τ is $E_u^\tau = m \cdot g \cdot h_u^\tau$, where m is the mass of the drone, g is the acceleration of gravity and h_u^τ is the altitude of the drone at time τ. The initial amount of energy is equal to E_0 and it is the same for all the drones. The movement of the drones is restricted by a minimum altitude h_{min} and a maximum altitude h_{max}.

The drones are, also, equipped with a wireless network interface in order to communicate with other drones. The communication range is R. We assume that the communication energy cost is negligible compared to the flying cost.

2.1. Mathematical model

In order to describe the proposed mathematical model to represent the optimal flying drones location problem, it is useful to introduce the following notations and definition.

Let U denote the set of available drones. It is assumed that each target $t_i \in T$ is characterized by its coordinates (X_{t_i}, Y_{t_i}). Given a drone u located in (x, y, h) and the target t_i, $D_{t_i}^{uxy} = \sqrt{(X_{t_i} - x)^2 + (Y_{t_i} - y)^2}$ represents the distance between u and t_i when $h = 0$.

Each drone u has a visibility that is represented by a circle in the plane with radius r^{h_u}. The higher the component h_u, the longer the radius. The decision variables are defined in what follows.

$$\delta_{xyh}^u = \begin{cases} 1, & \text{if the drone } u \text{ is located at} \\ & \text{the point of coordinates } (x, y, h) \\ 0, & otherwise \end{cases} \qquad \gamma_{t_i}^u = \begin{cases} 1, & \text{if the target } t_i \text{ is observed by the drone } u \\ 0, & otherwise \end{cases} \tag{1}$$

We assume that the value of h belongs to the interval h_{min}, h_{max} and the projection of the flying region is represented by rectangular with a length of x_{max} and a width of y_{max}. In addition, a time window $[\tau_{min}^{t_i}, \tau_{max}^{t_i}]$

is associated with each target $t_i \in T$. This means that the target t_i, initially located at the point of coordinates (X_{t_i}, Y_{t_i}), must be observed in the time range defined by the corresponding time window.

Each target changes its position during the time. In order to represent the dynamism of the system, starting from (X_{t_i}, Y_{t_i}), a sequence of coordinates C_i is associated with each target t_i. We assume that $C_i = \left\lceil \frac{\tau^{t_i}_{max} - \tau^{t_i}_{min}}{\Delta\tau} \right\rceil$, where $\Delta\tau$ is the time interval in which a new position of the target t_i is reached.

In order to represent the movement, target t_i is replaced with $|C_i|$ copies. A time window is associated with each copy t_i^j, $j = 1, \dots |C_i|$ of target t_i. In particular, $[\tau^{t_i}_{min}, \tau^{t_i}_{min} + \Delta\tau]$ is the time window associated with the first copy of t_i, whereas the time window associated with the copy t_i^{j+1}, $j = 1, \dots, |C_i| - 1$, is defined as $[\tau^{t_i^j}_{max}, \tau^{t_i^j}_{max} + \Delta\tau]$. Of course, if $\tau^{t_i^{|C_i|}}_{max} > \tau^{t_i}_{max}$, then $\tau^{t_i^j}_{max} = \tau^{t_i}_{max}$. The copies of each target are stored in the set $\dot{T}$. Let τ^u_{start} and τ^u_{end} be the initial and final time of observation of drone u, respectively. The mathematical formulation of the optimal flying drones location problem is reported in what follows.

$$\min \textstyle\sum_{u \in U} (\tau^u_{end} - \tau^u_{start}) \left(\alpha \sum_{(x,y,h)} h \delta^u_{xyh} \right) \tag{2}$$

$s.t.$

$$\textstyle\sum_{x=1}^{x_{max}} \sum_{(x,y,h)} \delta^u_{xyh} \leq 1 \quad \forall u \in U \tag{3}$$

$$\gamma^u_{t_i^j} \leq \textstyle\sum_{(x,y,h)} \delta^u_{xyh} \left(\frac{r^{h_u}}{D^{uxy}_{t_i^j}} \right) \forall u \in U, t_i^j \in \dot{T} \tag{4}$$

$$\textstyle\sum_{u \in U} \gamma^u_{t_i^j} = 1 \quad \forall t_i^j \in \dot{T} \tag{5}$$

$$\tau^u_{start} \leq \tau^{t_i^j}_{min} \gamma^u_{t_i^j} \ \forall u \in U, t_i^j \in \dot{T}, \ \tau^u_{end} \geq \tau^{t_i^j}_{max} \gamma^u_{t_i^j} \ \forall u \in U, t_i^j \in \dot{T} \tag{6}$$

$$\delta^u_{xyh} \in \{0,1\}, \ \forall (x,y,h) \tag{7}$$

$$1 \leq x \leq x_{max}, \ 1 \leq y \leq y_{max}, \ h_{min} \leq h \leq h_{max}, u \in U;$$

$$\gamma^u_{t_i^j} \in \{0,1\}, \ \forall t_i^j \in \dot{T}, u \in U; \tag{8}$$

$$\tau_u, \tau^u_{start}, \tau^u_{end} \in \mathbb{R}, \forall u \in U \tag{9}$$

The objective function (2), to be minimized, represents the total energy consumption. Constraints (3) ensure that the drone u is located in at most one position. Conditions (4) are used to set the value of variable $\gamma^u_{t_i^j}$. In particular, if the radius is less than the distance, then $\gamma^u_{t_i^j}$ takes a value equal to 0, otherwise, the variable can assume a value equal to either 0 or 1. Constraints (5) ensure that each target is observed by exactly one drone. Conditions (6) define the initial and final time of observation of drone u, respectively, whereas constraints (7) – (9) specify the domain of the decision variables.

It is worth observing that the proposed mathematical model (2) - (9) assumes that an infinite number of drones is available. In particular, it is assumed that when a drone moves to another position, it is replaced by another one. This hypothesis allows to simplify the model and it does not influence the solution, since the objective is the minimization of the total energy consumption.

Even though the effective number of drones, with a given initial energy, needed to observe all targets does not represent an output of the proposed model, it is possible, starting from the optimal solution, to determine an estimate of this number. It is worth observing that the optimal flying drones location problem is formulated as a binary optimization model, characterized by a high number of variables and constraints. Thus, it is not possible to determine the optimal solution of the model (2) - (9), in a reasonable amount of time. Since the solution process is very time consuming even for small instances, the use of heuristic approach, that allows to determine feasible solution, is mandatory.

3. Mobile target tracking solutions

3.1. Localized Altitude Scheduler

The Localized Altitude Scheduler (LAS) provides the drones with the required functions in order to compute the minimum possible altitude based on (a) the position of the targets that each drone currently

Algorithm 1: A round of LAS

require: $T \neq \emptyset$, $ACTIVE \neq \emptyset$

foreach *drone* $u \in ACTIVE$ **do**
- detect events of T covered by u;
- add detected events in $CVRD_u$

foreach *drone* $u \in ACTIVE$ **do**
- $timestamp_u = \rho + \frac{h_{max}}{h_u}$;
- send $timestamp_u$ to 1-hop neighbors along with the position and event details;

foreach *drone* $u \in ACTIVE$ **do**
- **foreach** *msg received from neighbor* u' **do**
  - **if** $(timestamp_u < timestamp_{u'})$ & $(\frac{dist(u,u') + h_u \tan\theta' + h_{u'} \tan\theta'}{2\tan\theta'} < h_{max})$ & $((h_u + h_{u'}) > \frac{dist(u,u')}{2\tan\theta'})$ **then** add events of $CVRD_{u'}$ in $CVRD_u$;
- **foreach** *event* $e \in CVRD_{u'}$ **do**
  - **if** $(e \in CVRD_u)$ & $(timestamp_u >= timestamp_{u'})$ & $(dist(u,u') < (h_u \tan\theta' + h_{u'} \tan\theta'))$ **then** $CVRD_u = CVRD_u - \{e\}$;

foreach *drone* $u \in ACTIVE$ **do**
- **if** $CVRD_u \neq \emptyset$ **then**
  - compute new altitude h'_u;
  - **while** $h' > h_{max}$ **do**
    - find the most distant event t in $CVRD_u$;
    - send a new drone request to cover t;
    - $ACTIVE = ACTIVE \cup \{u''\}$;
    - compute new altitude h'_u;
  - update position;
- **else**
  - $ACTIVE = ACTIVE - \{u\}$;

covers, (b) the position of the targets that the neighboring drones cover, and (c) the minimum and the maximum allowed altitude.

The algorithm considers that at any time τ all the drones are able to estimate their position and detect the targets underneath. The procedure is divided in rounds and in each round (see Alg. 1) each drone decides its state. There are two kinds of state. In the *active* state the drone monitors at least one event and communicates with other drones. In the *inactive* state the drone abandons coverage and remains on the ground when the targets it covers can be covered by neighboring drones. This process is explained later in the text.

Each round starts by detecting the events that each active drone covers. The initial number of active drones depends on the initial placement. The initial placement of the robots can be done either randomly or by placing them manually. Since the events have been detected and their position has been identified, each active drone computes a timestamp taking into account its current altitude and a random value ρ. ρ is used to avoid having two drones with same timestamp.

The timestamp of each active drone is broadcasted and the received timestamps are used by a drone to decide if it will give up monitoring certain targets in case of a merging. A merging takes place due to two reasons: (a) two neighboring drones with low altitude are very close to each other such as the events they cover together, can now be covered using only one of the drones, and (b) one or more events covered by a drone with low altitude come into the range of a drone with higher altitude. In the first case, a merging takes place if the total energy consumption of the two (or more) drones before the merging is more than the energy consumption of the single active drone after the merging. This check is done by the drone with the lowest timestamp when the involving drones come close to each other.

In the first case, the left part of formula (10) makes sure that the new drone altitude will not exceed the highest possible altitude, while the right part compares the energy consumption before and after the merging. $dist(u, u')$ represents the current distance between the centers of the flat surface of the cones and $\theta' = \theta/2$.

$$\frac{dist(u,u') + h_u \tan\theta' + h_{u'} \tan\theta'}{2\tan\theta'} < h_{max}, \qquad h_u + h_{u'} > \frac{dist(u,u')}{2\tan\theta'} \tag{10}$$

In the second case, $dist(u, u') < (h_u \tan\theta' + h_{u'} \tan\theta')$ checks if u has been inserted into u' range.

Since the robots have exchanged timestamps, they continue with the computation of their new position.

Each active drone u maintains the information about the targets it covers in $CVRD_u$. It can calculate the final altitude by computing the radius of the circle that includes all the targets in $CVRD_u$. This computation can be done using a smallest enclosing circle (SEC) method or another suboptimal solution.

The computation complexity of the SEC is at least $O(\nu)$ [10], where ν is the number of events that take part in the computation. In scenarios where a drone may cover several events, the altitude computation must be as fast as possible, otherwise other operations such as the target detection would be delayed causing a possible appearance of uncovered events. For that reason suboptimal solutions with lower computation cost can be used.

Moreover, depending on the speed of the targets and how often the drones detect the events, the actual altitude of the drones must be higher than the one computed by the SEC. The final drone altitude is computed using the following formula:

$$h_u = h'_u \tan\theta' + \upsilon_{max} dt \tag{11}$$

where dt is time elapsed between two target detections and h'_u is the altitude calculated by the SEC algorithm (or by another suboptimal solution). This formula ensures that the events in $CVRD_u$ will still be covered until the next detection.

In some scenarios, as the targets go away the one with each other, the altitude of a drone may reach its final value. In this case, the drone must decide which of the drones will follow, abandoning the monitoring of the most distant event it covers and decreasing the altitude. Using a broadcast it demands an assistant from the closest base to cover this event. Since the drones have the capability of moving fast, we assume that the drone-assistant arrives during the same round.

A drone running LAS uses two types of messages to communicate with other drones. The first type is a broadcasting packet containing information about the timestamp, its initial position and the events it monitors. The second type is used to inform other drones that an abandoned event must be covered making LAS as much tolerant as possible in presence of message failures. A possible packet loss of the first message type does not affect the monitoring of the events since in the worst case it could only lead to a double or triple covering of some of the targets. On the other hand, a packet loss of the second message type leads to uncovered events if the abandoned event is not covered by any other drone at that particular moment.

Moreover, LAS does not require any synchronization between the drones since each drone can act independently from the others. Non-synchronized neighboring drones do not lead to the appearance of uncovered targets, since they abandon targets only if these particular targets are already covered by another drone.

3.2. Centralized Altitude Scheduler

The centralized altitude scheduler (CAS) works as a greedy heuristic and it assumes that the position of the events and their movement is known in advance, so it is difficult to be used in practice. The aim of the algorithm is to minimize the consumption by keeping the active drones in low altitude and avoiding overlappings (i.e. two or more drones covering same targets). Note that it does not always mean that CAS will use the minimum possible number of drones.

The algorithm starts by allowing a single drone to cover as many events as possible. When it reaches the maximum possible altitude, another drone is used to cover the rest of the events. The new drone starts from the most distant uncovered event and it avoids covering the events that the previous drones have covered. The process terminates when all the events have been covered.

The complexity of the approach mainly depends on the complexity of the chosen SEC algorithm. Since the lowest complexity of computing it is $O(n)$, the overall complexity of CAS is $O(n^2)$.

4. Evaluation and discussion of the results

In this section we simulate the proposed algorithms and we compute the total energy consumption and the average number of drones used throughout the monitoring process. We present results using two versions of LAS; the first version uses the average of the coordinates of the events in order to compute the drone position, while the second one uses the SEC solution. We name these two versions “LAS” and “LAS-sec” respectively. “Static” uses the minimum possible number of drones to cover all the terrain area by placing the robots based on a triangular grid. In this case, all the drones are in the same altitude, while the altitude varies depending on the terrain size. Since the problem is new and there is no other similar algorithm in the literature, we compare LAS against CAS and Static approaches.

We assess the algorithms in two scenarios. In the first scenario, we vary the terrain size keeping constant the number of events in the field (100 events), while in the second scenario we keep fixed the terrain size (40K m^2), varying the number of events. In the first case, we evaluate the behavior of the approaches when few or many drones are used, while in the second case we assess the event exchange behavior when few or many events exist. For each scenario, we use 50 topologies with random and uniform initial event deployment and we compute the average values of these 50 topologies. The 95% confidence intervals are shown in each figure. Each drone has a communication range of 150m, while the mass of the drone is 10Kg, g is 10m/sec^2, and the minimum and the maximum allowed altitudes h_{min} and h_{max} are equal to 10m and 100m respectively. The camera angle θ is set 60° which corresponds to a normal camera lens. We assume that the starting position of the events is known, so the drones are initially placed according to this position. If this information is not known, the "Static" model can be used to cover all the area.

Concerning the events, we evaluate our approaches using three mobility models. In the first model, each event is attracted by a certain point in the terrain (Attractors model). The number of the attractors is 5 and their position is fixed. In the second model, the events are moved with a random speed towards a point that they also choose randomly (RWP model). In the third case, the events are moved completely randomly without stopping at any point (Random model). In all the cases, the maximum speed of an event is 1m/sec, each simulation lasts 500secs, and the position of the drones is updated every second. Since the events might reach the final destination before the finish time, we mostly present evaluation findings for the random mobility model. Otherwise, the results would be affected by the time the events stayed static.

4.1. Simulation results

The results of Figure 1 show that the performance gap between LAS and CAS is small. This performance difference is even lower when the SEC solution is used for the computation of the position of the drones. CAS presents the best performance activating though more drones as it is shown in Figure 2. On the contrary, the static approach presents similar results when the terrain size is low, but the energy consumption increases a lot for large terrain sizes.

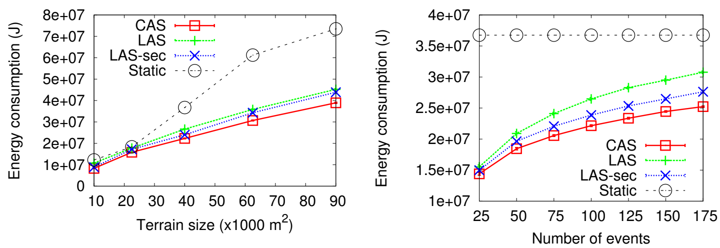

Fig. 1. Total energy consumption in relation with the terrain size (left) and the number of events (right)

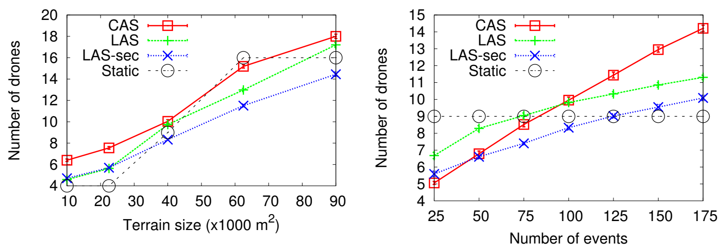

Fig. 2. Average number of active drones in relation with the terrain size (left) and the number of events (right)

In the next experiment, we vary the detection frequency (i.e. the time elapsed between two successive target detections) and we measure the total energy consumption for the two scenarios. According to Formula (11) the final altitude of a drone depends on the detection frequency, which practically means that the active drones should fly higher in order to still be able to detect the mobile events. The results are presented in Figure 3 and show that for dense scenarios and high detection frequency (e.g. 10 secs), the total energy consumption is slightly higher than that of the static approach. For lower detection frequencies LAS still performs better than Static. We must mention though that LAS uses the average of the coordinates of the events to compute the position of the drones. On the contrary, the energy consumption could be decreased using a SEC method.

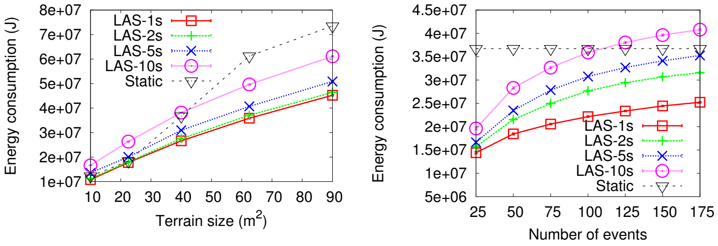

Fig. 3. Total energy consumption in relation with the terrain size (left) & the number of events (right) for different detection frequencies

Figure 4 illustrates the performance of LAS in presence of transmission or reception errors due to noisy environment, channel congestion etc.. We assume a 50% failure each time a drones tries to communicate with other drones. When failures occur, one or more events may be left uncovered as it is explained in Section 1. Despite the huge failure probability, the percentage of uncovered events throughout the monitoring time was 2-3% for the dense scenarios and up to 10% for the sparse scenarios (left figure). Since in case of failures the number of active drones is lower, the energy consumption is slightly reduced (right figure).

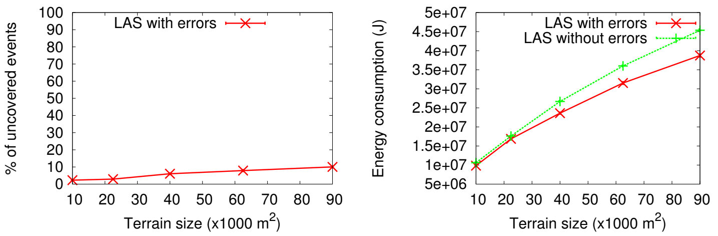

Fig. 4. Percentage of uncovered events (left) and total energy consumption (right) in relation with the terrain size

Figure 5 represents the energy consumption of the active drones at any time for the three mobility models. When the random mobility model is used, the energy consumption reasonably fluctuates over time, but in case of “attractors” or “Random” mobility models LAS is able to gradually decrease the energy consumption by activating less drones since the events are getting closer to the destination. This case is, also, illustrated in Figure 6. The events are initially scattered all over the terrain while they stop moving when they reach one of the attractors. The final rounds include four active drones, three with low altitude and one with higher altitude covering two attractor sites[1].

5. Conclusion and future work

In this paper we dealt with the power efficient scheduling problem of a fleet of robots that monitor events on the ground. We presented LAS, a localized algorithm that controls mobility of flying robots.

[1]Animated depictions produced by our simulation software can be found at `http://uav-scheduling.gforge.inria.fr/`

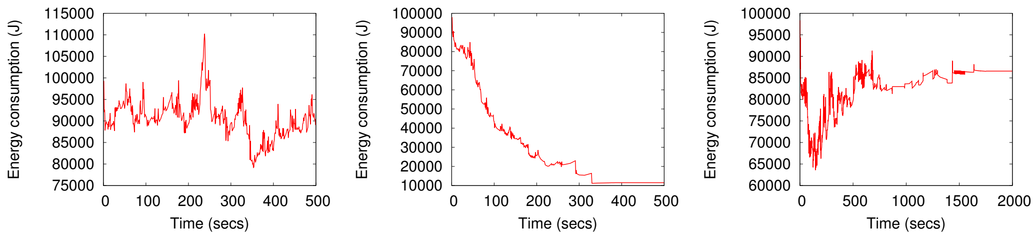

Fig. 5. Energy consumption of LAS throughout the monitoring process for a scenario with 100 events, 90K m^2 terrain size, random mobility model (left), attractors mobility model (middle) and RWP mobility model (right)

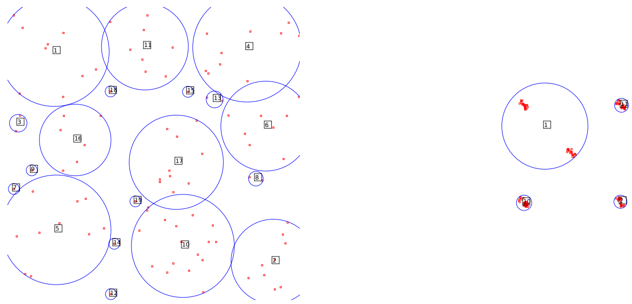

Fig. 6. A 2-dimensional depiction of the initial and the final position of the events (dots) and the drones (squares with numbers) for a scenario with 100 events, 90K m^2 terrain size and attractors mobility model

The drones adjust their altitude to cover more or less targets at each time. This adjustment is achieved by communicating with the neighboring drones and it leads to an up to 150% energy conservation compared to the case where the drones are statically placed. Our future work involves the use of a metric to control the detection probability of each event as well as the use of other mobility models.

Acknowledgements

This work is partially funded by the French National Research Agency (ANR) under the ANR VERSO RESCUE project, grant number ANR-10-VERS-003.

References

[1] H. Chen, X. min Wang, Y. Li, A survey of autonomous control for uav, in: Artificial Intelligence and Computational Intelligence, 2009. AICI '09. International Conference on, Vol. 2, 2009, pp. 267 –271.

[2] P. Zhan, D. Casbeer, A. Swindlehurst, A centralized control algorithm for target tracking with uavs, in: Signals, Systems and Computers, 2005. Conference Record of the Thirty-Ninth Asilomar Conference on, 2005, pp. 1148 – 1152.

[3] M. Mallick, Geolocation using video sensor measurements, in: Information Fusion, 2007 10th International Conference on, 2007, pp. 1 –8.

[4] P. Theodorakopoulos, S. Lacroix, A strategy for tracking a ground target with a uav, in: Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on, 2008, pp. 1254 –1259.

[5] S. Quintero, F. Papi, D. Klein, L. Chisci, J. Hespanha, Optimal uav coordination for target tracking using dynamic programming, in: Decision and Control (CDC), 2010 49th IEEE Conference on, 2010, pp. 4541 –4546.

[6] V. Dobrokhodov, I. Kaminer, K. Jones, R. Ghabcheloo, Vision-based tracking and motion estimation for moving targets using small uavs, in: American Control Conference, 2006, 2006, p. 6 pp.

[7] A. Shimada, M. Fujita, Takeoff and landing control using force sensor by electrically-powered helicopters, in: Advanced Motion Control, 2006. 9th IEEE International Workshop on, 2006, pp. 62 –65.

[8] Y. Kubota, Y. Iwatani, Dependable takeoff and landing control of a small-scale helicopter with a wireless camera, in: Robotics and Biomimetics (ROBIO), 2011 IEEE International Conference on, 2011, pp. 1279 –1284.

[9] K.-H. Hsia, S.-F. Lien, J.-P. Su, Height estimation via stereo vision system for unmanned helicopter autonomous landing, in: Computer Communication Control and Automation (3CA), 2010 International Symposium on, Vol. 2, 2010, pp. 257 –260.

[10] S. Xu, R. Freund, J. Sun, Solution methodologies for the smallest enclosing circle problem, Computational Optimization and Applications 25 (2003) 283–292.