Proceedings of the Twenty-Fifth International Conference on Automated Planning and Scheduling

An Online Replanning Approach for Crop Fields Mapping with Autonomous UAVs

Alexandre Albore,[†‡] **Nathalie Peyrard,**[†] **Régis Sabbadin,**[†] and **Florent Teichteil-Königsbuch**[‡]
† INRA UR 875 (MIA), chemin de Borderouge, 31326 Castanet-Tolosan, France - *name.surname*@toulouse.inra.fr
‡ ONERA (DCSD) - 2 avenue É. Belin, 31000 Toulouse, France - *name.surname*@onera.fr

Abstract

For managing production at the scale of crop fields, maps of plant pests are used to support farmer decisions. Such maps are costly to obtain since they require intensive surveys in the field, most of the time performed by human annotators or with human-controlled Unmanned Aerial Vehicles (UAVs). In this paper, we look at the next challenge from an AI planning point of view: flying fully autonomous UAVs equipped with online sequential decision-making capabilities for pests sampling and mapping in crop fields. Following existing work, we use a Markov Random Field framework to represent knowledge about the uncertain map and its quality, in order to compute an optimised pest-sampling policy. Since this planning problem is PSPACE hard, thus too hard to be exactly solved either offline or online, we propose an approach interleaving planning and execution, inspired by recent works on fault-tolerant planning. From past observations at a given time step, we compute a full *plan* consisting in a sequence of observed locations *and* expected observations till the end of the pest-sampling phase. The plan is then applied until the number of actual observations that differ from expected ones exceeds a given threshold, which triggers a new replanning episode. Our planning method favourably compares on the problem of weed map construction against an existing greedy approach – the only one working online – while adding the advantage of being adapted to autonomous UAVs' flying time constraints.

Introduction

Flying UAVs to help farmers map crop fields. An important tool in precision agriculture for supporting the management of production in crop fields is a map of pest abundance spatial distribution. However, such maps are costly to obtain since they require intensive pest sampling in the field, until now mainly performed by human annotators; under these conditions, the whole field cannot generally be sampled and a complete abundance map is estimated from samples in sub areas of the field. Remote sensing tools are emerging as a promising alternative due to their flexibility to gather information on large areas. So far, the primary remote platforms for collecting images in agriculture have been piloted aircraft and satellites. Such platforms are giving way to Unmanned Aerial Vehicles (UAVs) that provide a better spatial and temporal resolution for the image analysis at a lower cost, and offer an ideal point of view for the acquisition of ground-based data. Moreover, UAVs can operate below cloud covers – which is impossible for higher altitude aircraft and satellites – and can be deployed quickly and repeatedly, which permits an almost continuous monitoring of an area. Even with UAVs, a sampling strategy has to be determined because of the limited flight time, and the extension of the area to map, generally too big for an exhaustive monitoring.

Autonomous systems are well-suited to exploit dynamic information from images and for in-flight optimisation of navigation between areas to sample. The same operations cannot be easily performed by piloted UAVs, as computations and heuristic evaluations to choose the best trajectory and altitude for optimal pest sampling are generally non trivial. So, the next challenge is to use completely automated UAVs, with on-board computation capabilities.

On the other side, autonomous UAVs should be deployed on demand, without heavy logistics as side effects of autonomy capabilities; this includes expensive ground computational units or hours-long night computations to optimise exploration strategies before the flight. Therefore, strategy optimisation must be conducted on-board, during the flight, at a low computational cost: farmers actually expect automated techniques to be as easy as possible, with UAVs ready to be used with no external computations nor solution strategies uploaded to the on-board memory, that may force to foresee weather conditions several days before the flights.

A decision-theoretic approach based on Markov random fields. Adaptive sampling techniques have been developed in the context of manual sampling done by humans, with the purpose of mapping invasive species or weeds in large areas (Peyrard et al. 2013; Bonneau et al. 2014). These approaches rely on a Markov Random Field model (MRF) (Geman and Geman 1984) of the abundance map to estimate, and on methods for sequential decision-making under uncertainty. A map is divided into plots of ground (or sites) and the MRF enables to model marginal abundance distributions at each site, which are updated after each new sampled observation.

In (Peyrard et al. 2013), observations result from sampled sites chosen *greedily*, without considering the future sampling steps and with no consideration of total sampling bud-

get. As a consequence, this adaptive sampling technique is rather fast, but does not optimise resources such as remaining flying time, which in turn have an impact on the quality of the resulting map.

(Bonneau et al. 2014) improved the previous approach by including in the adaptive sampling design the full sampling horizon and the sampling budget. By casting this problem into the framework of Markov Decision Processes (MDPs), they derived a Reinforcement Learning approach to construct parametric sampling policies. The drawback of this approach is the very long off-line computation time required to compute the parameters of the sampling policy (can be several hours). Therefore, this approach does not match our needs for on-line and on-board computations as required in the context of on-demand UAV-based pest sampling with light logistics.

Moving to a decoupled allocation and replanning approach. Our objective, in this paper, is to propose a non-greedy approach for sampling in MRF, improving sampling quality compared to the greedy approach of (Peyrard et al. 2013), but without requiring as much computation time and efforts as the RL approach of (Bonneau et al. 2014). In order to achieve this challenging time-constrained objective, we propose to dissociate the problem of selecting the observation sites from the one of planning their visiting order.

Therefore, we model the task of finding an efficient sites visiting sequence as a classical planning problem, and use a planner to synthesise the sampling strategy based on the observed values on the ground, while guaranteeing that the plan is always executable by the UAV given its remaining flight time. Concretely, from past observations history at a given time step, we compute a full *plan* till the end of the sampling phase, considering that moving and information gathering actions differ in cost. This plan consists in a sequence of locations *and* expected observations. The plan is then executed until the number of actual observations that differ from expected ones exceeds a given threshold. Only then, a new plan is computed, then executed until its environmental assumptions become *too much* violated. This replanning approach is close in spirit to *fault-tolerant* planning (Domshlak 2013) and *planning under assumptions* (Albore and Bertoli 2006), with the difference that in this problem, no execution dead ends can occur.

Note that the planning model here is still not a variant of the well-known travelling salesman problem, even if there are some points in common; here, the information obtained when sampling a plot has the side effect of reducing the expected rewards from neighbouring plots. This influence is to be considered when producing a visiting order, and a planner is more suitable to heuristically consider such causal links: namely, observing a given site prevents from exploring nearby locations because it may not substantially improve the knowledge of the map.

In the following, we first recall the modelling of the problem of optimal pest sampling for mapping in MRF as described in (Bonneau et al. 2014) and explain why the solution based on MDPs is not suitable for online mapping with UAVs. We then present our original approach based on interleaving planning and execution. Its performance is empirically illustrated on a problem of weeds sampling in crop fields, and compares favourably, in terms of quality and resource consumption, to the greedy approach, which is the only existing online solution to this problem.

Modelling UAV-based sampling as a sequential decision-making problem under uncertainty

We will consider that the crop field to map is divided into a regular grid of N plots of small area. Observing a plot provides the weed abundance there (discretised in K classes) and we assume no measurement error. It is impossible to use the UAV to acquire an observation in each site, due to limited battery capacity. Only a sample of the total plots can be observed, and from this sample we want to provide a full map of the field by estimating the value at unobserved sites. To do so, we consider the problem of designing an adaptive sequence of sampled plots (an adaptive plan) that maximises the quality of the estimated map, reconstructed from the gathered observations, under UAV's physical constraints. An adaptive plan implies that the sequence of plots is not defined beforehand, and the next site where to sample may depend on the history of previous observations (positions and values) and it is determined dynamically. We follow the work of (Peyrard et al. 2013) where MRF are used to model weed abundance maps distributions, and the definitions of estimation and quality are based on the remaining uncertainty in the estimated complete map.

MRF modelling of abundance maps

The MRF model for abundance map is defined as follows. To each site $i \in V = \{1, \dots, N\}$ of the field is attached a discrete random variable X_i with domain $D = \{1 \dots K\}$, where K is the number of abundance classes. The joint distribution of the whole map $X = (X_1, \dots, X_N)$ is assumed to be expressed as a pairwise MRF: $\forall x \in D^N$,

$$\mathbb{P}(X = x) = \frac{1}{Z} \prod_{i=1}^{N} f_i(x_i) \prod_{(i,j) \in E} f_{i,j}(x_i, x_j) \qquad (1)$$

The set E is the set of all pairs of order 1 neighbours in the grid of sites and Z is a normalising constant. The f_i and $f_{i,j}$ are non negative functions called respectively order 1 and order 2 potential functions. Roughly speaking, the order 1 potential functions weight the relative proportions of the K abundance classes while the order 2 potential functions encode spatial correlation between abundance values at different sites. The choice of an appropriate MRF model amounts to the choice of these potential functions. We will provide an example of such a choice in the case of weeds maps.

The problem of optimal adaptive sampling in MRF with an objective of map reconstruction has been modelled in (Bonneau, Peyrard, and Sabbadin 2012) as the problem of finding a policy tree that optimises a given criterion. It relies on the following elements.

Reconstruction. When an output x_A is available (for A, a subset of V), the Maximum Posterior Marginals (MPM)

criterion, classically used in image analysis, is used to derive an estimator x^* of the hidden map x:

$$x^* = \Big\{ x_i^* \;\big|\; i \in V, \quad x_i^* = \underset{x_i \in D}{\operatorname{argmax}}\, \mathbb{P}(x_i \mid x_A) \Big\}.$$

Note that one of the main difficulties when using the MPM criterion is to compute the values of all conditional marginals $\mathbb{P}(X_i = k | X_A = x_A)$. Exact inference algorithms, like junction tree and Monte Carlo methods, calculate exact marginal probability distributions. We tested such approaches, with the conclusion that only approximate computation is adapted to our application: the junction tree algorithm requires too much memory space, while Monte Carlo methods are overly time consuming. The classical Loopy Belief-Propagation (LBP) algorithm (Murphy, Weiss, and Jordan 1999) is often used to approximate inference in MRF. LBP provides no theoretical guarantees on convergence, even so it is largely used for its good behaviour in practice and its rapidity. Note that guaranteed approximations are NP-hard to obtain for marginals computation in MRF (Roth 1996).

Adaptive sampling policy. In adaptive sampling, the sample A is chosen sequentially. The sampling sequence is divided into H steps. $A^h \subseteq V$ is the set of sites explored at step $h \in \{1, \dots, H\}$ and x_{A^h} is the sample output at step h. The choice of sample A^h depends on the previous samples and outputs. An adaptive sampling policy $\delta = (\delta^1, \dots, \delta^H)$ is then defined by an initial sample A^1 and functions δ^h specifying the sample chosen at step $h \geq 2$, depending on the results of the previous steps: $\delta^h((A^1, x_{A^1}), \dots, (A^{h-1}, x_{A^{h-1}})) = A^h$.

The relevant information in a trajectory followed when applying δ can be summarised by the pair (A, x_A), where $A = \bigcup_h A^h$.

Sample cost. For UAV based sampling, the cost of a trajectory will be directly related to the energy consumption of the UAV. In the case where one site is sampled at a time, it is defined as:

$$c\big((a^1, x_{a^1}) \dots, (a^H, x_{a^H})\big) = c_1(a^1, x_{a^1}) + \\ + \sum_{h=1}^{H-1} c_2(a^h, a^{h+1}, x_{a^{h+1}}),$$

with $c_2(a^h, a^{h+1}, x_{a^{h+1}}) = d(a^h, a^{h+1}) + c_1(a^{h+1}, x_{a^{h+1}})$ and $c_1(a, x_a)$ is the energy needed to tell that site a is in state x_a (this energy consumption may depend on the observation), with $d(a^h, a^{h+1})$ the energy required to fly from site a^h to site a^{h+1}. This energy is related to the distance, the wind direction and force, etc.

Quality of a sampling policy. The quality of a policy δ is measured as the expected quality of the estimator x^* that can be obtained from δ. In practice, we first define the quality of a trajectory $((A_h, x_{A_h}))_{h=1..H}$ as a function of (A, x_A), where $A = \bigcup_h A_h$:

$$U\big(A, x_A\big) = \sum_{i \in V} \Big[\max_{x_i \in D} \big\{ \mathbb{P}(x_i \mid x_A) \big\} \Big]. \qquad (2)$$

$U\big(A, x_A\big)$ can be interpreted as the expectation of the number of well-classified sites, when allocating their values to the modes of the marginals $\mathbb{P}(x_i \mid x_A)$.

We note τ_δ the set of all possible trajectories that can be followed by executing δ, which represents a *policy tree* (see Fig. 1). The quality of a sampling policy δ is then defined as an expectation over all possible trajectories:

$$V(\delta) = \sum_{((A_h, x_{A_h}))_h \in \tau_\delta} \mathbb{P}\big(x_A\big) \cdot U\big(A, x_A\big).$$

The problem of optimal adaptive sampling is to find the policy of highest quality under sampling budget constraint :

$$\delta^* = \underset{c(\delta) \leq B}{\operatorname{argmax}}\, V(\delta). \qquad (3)$$

Here, $c(\delta)$ is defined as the maximum of the costs of all trajectories in τ_δ, and B is a fixed *budget* for sampling. In the UAV case, B is the battery total energy and a policy is admissible only if none of the trajectories it can generate use more energy than available. Apart from the cost constraint, our problem is similar to a Partially Observable Markov Decision Process with terminal rewards $U(A, x_A)$, which generally prevents to solve it online on-board the UAV for limited-CPU and time reasons. The approximate solution methods we will describe next allow to verify, on-line, that the policy currently generated is admissible.

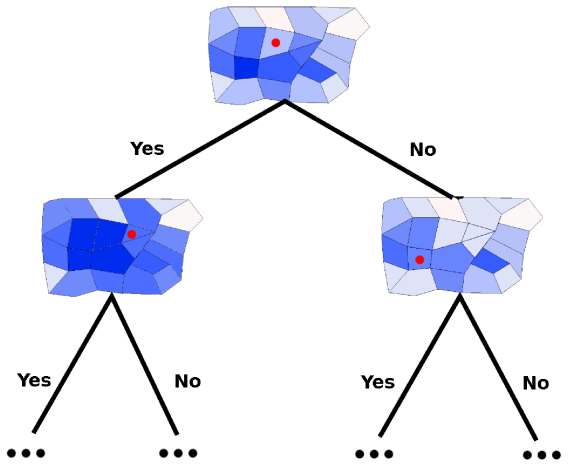

Figure 1: Example of a policy tree for a binary MRF of presence/absence map. The red dot is the observed site, and Yes/No branches are the possible observations. A site colour encodes the value of the marginal probability of presence, that is updated after each new observation. A plan is a path in this tree representation.

Approximate solution methods

The sampling policy optimisation problem defined above is too hard to solve exactly for two independent reasons:

- (Bonneau, Peyrard, and Sabbadin 2012) have shown that it was PSPACE-complete to determine whether there exists an adaptive sampling policy whose utility is above a given threshold. Furthermore, representing such a policy tree takes exponential space in the problem description.

- Computing the MPM for a given sampling observation is #P-complete and NP-hard to approximate (Roth 1996). This computational cost has to be paid by any planning method tackling this problem, which means that we should opt for an online approach visiting as few states and evaluating as few actions as possible.

Several approaches have been proposed to approximately solve this problem, but none seems to fit the needs of our application-oriented framework. The *greedy* approach mentioned before (Peyrard et al. 2013) fails in considering neither future sampling steps nor the available sampling budget. A *reinforcement learning (RL)* type approach (Barto, Sutton, and Watkins 1989; Bonneau, Peyrard, and Sabbadin 2012), together with a dedicated value-function approximation method, would require excessive "off-line" computation resources to solve the RL problem.

A replanning-based approach for efficient pest sampling and field mapping

The original approach proposed here to solve the optimisation problem stated above, uses a replanning framework to generate a pest sampling strategy – in fact a partial MDP policy – synthesised by means of several calls to a classical (deterministic) planner. There are mainly three flavours of this approach. The first variant is purely reactive (Yoon, Fern, and Givan 2007; Brafman and Shani 2012): it assumes that the environment is deterministic so that it generates a plan from the current state of the world which is thrown away and replaced by a new one as soon as its execution deviates from the expected state sequence. The second one relies on the same deterministic relaxation of the environment, but constructs a beam of plans that are merged into a partial policy (Teichteil-Königsbuch, Kuter, and Infantes 2010), or helps to optimise a Bellman-based policy (Yoon et al. 2010). These are not the best candidate variants for our problem: the first would use a static model of the field, either too optimistic or too generic, that wouldn't take into account the possible observations; the second, on the other side, would require to compute too many marginals for each plan, which would take too much time and need too much computational resource to be used online and on-board the UAV. The third flavour, which we will rely on next, generates a single plan from the current state, and executes it like the first variant, except that the representation of the environment is updated when the observations deriving from the execution of the plan deviate too much from the expected values. The tolerance to a certain amount of error between the representation used for planning and the real world is related to the idea of fault-tolerant planning (Domshlak 2013; Albore and Bertoli 2006), where a maximum number of unexpected effects can comply with the planning execution.

The planning problem of reconstructing a pest abundance spatial distribution map with an UAV is built and executed in a closed-loop fashion. First, we generate a set of n plots to sample[1] that maximise the *expected quality gain*, a quantity defined below and derived from the MRF model. Then, a planner is called to solve the task of finding a trajectory that minimises the navigation cost while visiting all the n sampling sites. The synthesised plan is then executed by the UAV, and the observations collected. We monitor the execution so to stop it and replan whenever the accumulated difference between the quality gain observed and the expected one is too high. In this case, we also update the expected quality gain for all the sites.

After a given set of observations, (A, x_A), we define, for each variable X_i of the MRF, the *expected quality gain* as an optimistic approximation of the increase of the updated utility $U(A \cup \{i\}, x_A, X_i) - U(A, x_A)$. For a variable X_i, we define the expected quality gain $\bar{q}(X_i, x_A)$ as:

$$\bar{q}(X_i, x_A) = \max_k \Big(\sum_{dist(i,j) \leq r} \max_{x_j} \mathbb{P}(x_j | X_i = k, x_A) \Big) + - \sum_{dist(i,j) \leq r} \max_{x_j} \mathbb{P}(x_j | x_A). \quad (4)$$

A Classical Planning Model for the navigation task

The navigation problem in an unknown field can be modelled as a deterministic planning model with action costs. Such model can be characterised as a tuple $\mathcal{S} = \langle S, s_0, S_G, A, f, c \rangle$ where S is a finite set of states, $s_0 \subseteq S$ is the initial state, S_G is the set of goal states, A is a set of actions with $A(s)$ denoting the actions in A that are applicable in the state s. An action a applicable in a state s changes the state to $s' = f(a, s)$, with $f : A \times S \to S$ the transition function. The cost function c for actions is $c : A \to \mathbb{R}_0^+$.

An action sequence $\pi = a_0, \ldots, a_n$ is applicable in a state s_0 if $a_i \in A(s_i)$, $0 \leq i \leq n$, and there exists a sequence of states $s_0, \ldots, s_{n+1}$, such that $s_{i+1} = f(a_i, s_i)$; in such a case we say that π achieves s_{n+1} when executed in s_0. π is a plan for P if it achieves a state g in G, when executed in the initial state. The cost of a plan π is $c(\pi) = \sum_{a \in \pi} c(a)$.

For the UAV's navigation problem, the planning space consists in states encoding an UAV's pose, and the status of the sites (observed/not observed). The initial state is given by the UAV's initial pose, while the goal is to have performed an observation in all the sites given in a set that maximises the expected quality gain. We consider two kinds of actions: *goto* actions move the agent between neighbour sites, and each *observation* action flags a plot and its neighbours in the field as observed. This causal relation between observed sites and their neighbours is dictated by the result of updating a site i in the MRF with an observation: the max marginal values of the neighbour sites vary more for sites closer to i. This implies that the quality gain expected from observing the value of a site is small if it is close to a plot that has already been observed.

Moving between two adjacent sites has unit cost, which corresponds to measuring the Manhattan distance for distant sites, while sampling has a slightly higher cost, as we

[1] We decide here to select a number n of plots depending on the UAV's autonomy, and on planning performances. Using all possible plots would reduce the problem to visit all the field's plots in a given order, which is practically unfeasible, while using only the best valuated site would end up being very close to the greedy adaptive sampling technique described above.

consider that stabilising the UAV to take a picture consumes more resource than flying between two adjacent plots.

Interleaving planning and execution The map reconstruction task is separated in two clear parts: (1) selection of observation sites; (2) search of a visiting order that optimises the flying time constraints of the UAV. This decoupling permits to observe and update the knowledge model within a robotic platform in real time while moving to the next site.

To guarantee that distance constraints (reflected in the flying time) are respected, the states that violate the given constraints are pruned from the search space, in a very similar way to what (Ivankovic et al. 2014) do using global numerical state constraints (but without considering them yet in the heuristic evaluation), or what the planner MBP (Bertoli et al. 2001) does with problem invariants coded as the verification of invariant properties in the NuSMV model checker (Cimatti et al. 2000). To track the distance flown, we use a numerical variable that is updated along with the state and which value is monitored at planning-time.

On top of this constraint, replanning occurs when the quality requirements are not met at execution-time. We illustrate this behaviour in the following pseudo-code:

Algorithm 1: Main (re)planning loop.

```
1  Function ReplanningLoop(s_0):
       Input: initial state s_0
2      s ⟵ s_0;
3      goals ⟵ ∅;
4      repeat
5          goals ⟵ bestPlots();
6          π ⟵ plan(s, goals);
7          s ⟵ execute(π, s);
8      until π = ∅;
```

Algorithm 2: Plan execution.

```
1  Function Execute(π, s'):
       Input: plan π, state s'
       Output: current state s
2      s ⟵ s';
3      o̅ ⟵ ∅;
4      foreach a in π do
5          s ⟵ f(a, s);
6          if isObservation(a) then
7              q̅ ⟵ updateMRF(sampleResult(a, s));
8              o̅ ⟵ sampleResult(a, s);
9              if q̅ > ε then        /* replanning condition */
10                 recalculateQGains(o̅);
11                 break;
12     return s;
```

Routine *Replanning-loop(init)* takes the initial state of the planning problem as input and obtains n sampling locations by selecting the sites with the biggest expected quality gain in the MRF at step 5. This list is set to be the goals of the planning problem: at step 6 the planner synthesises a plan consisting in *goto* and *observation* actions. Step 7 updates the current state with the outcome of executing the plan π. The loop repeats until all the sites are visited, or no plan is synthesisable under the given constraints (step 8).

Routine *Execute(π, s)* applies the actions a in the plan π (step 5), and updates the current state. If a is an *observation* action, then the evidence obtained by the observation in the real world is used to update the MRF (step 7) and the cumulated difference in expected quality gain $\bar{q}$, which must remain smaller than a fixed value ϵ, otherwise a replanning episode is triggered, and expected quality gains are recalculated from the past observation history (step 10).

Empirical evaluation of the platform

We illustrate our online replanning approach on the problem of sampling for weeds mapping in crop fields. Weeds compete with crop for resources and can be hosts for parasites and diseases, but they also play an important role in biodiversity conservation. To design novel management strategies to maintain weeds and local biodiversity, while controlling production, weeds maps are an essential support tool that help understanding weeds spatial distribution.

We implemented the previously described replanning algorithm within the Robot Operating System (ROS) framework (Quigley et al. 2009), a robotic meta-operating system that provides hardware abstraction, low-level device control, implementation of commonly-used functionality, message-passing between processes, and package management. The evaluation of marginals in the MRF and the planner are integrated on the same platform, taking advantage of the libDAI implementation of the LBP algorithm (Mooij 2010), and for the (re)planning loop, we used the Lightweight Automated Planning Toolkit (Ramirez, Lipovetzky, and Muise 2014) run with a Serialized Iterated Width algorithm (Lipovetzky and Geffner 2012), that we adapted as a ROS independent planning package.

We ran all empirical evaluations using the MORSE simulator for academic robotics (Echeverria et al. 2011), which enables to perform software architecture-in-the-loop (SAIL) realistic simulations, i.e. to test the exact same functional architecture as the one that will be implemented on-board the real UAV, but replacing the physical sensors and actuators by simulated data. Interestingly, we can feed simulated sensors like cameras with real data such as true images, meaning that the image analysis algorithm can deal with the same kind of images in the simulation as during the real flight. We get weeds abundance classes in the field plots from the standard semantic camera sensor of MORSE (cf. Fig. 2) set to 50mm focal length. From a planning point of view, SAIL simulations allow us to test the planning algorithm in very realistic conditions. The memory limit was set to 1GB and the time-out at 20 minutes, using an Intel Xeon CPU at 3.70GHz. The relation between the UAV sensors and actuators, and the ROS nodes is shown in Fig. 3.

The execution within ROS starts with i) the planner synthesising a plan, i.e. an ordered sequence of instructions for the UAV, ii) the next *waypoint* to reach is sent to the UAV, iii) GPS coordinates (the *pose*) are sent back to the planner node, so that if an observation site is reached, the relative data can be gathered from the *camera* and used both to up-

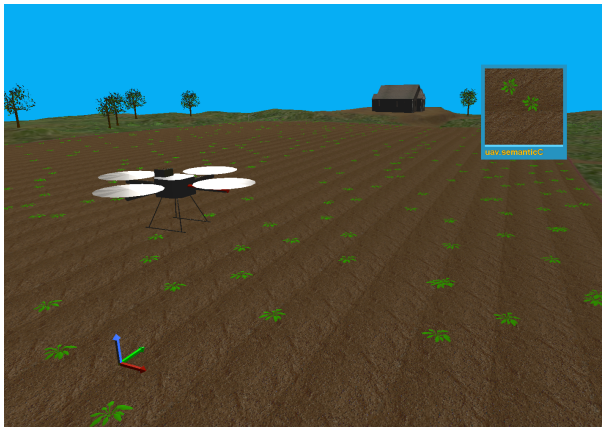

Figure 2: Platform simulation on MORSE. At the upper right corner, the UAV's semantic camera framing weeds.

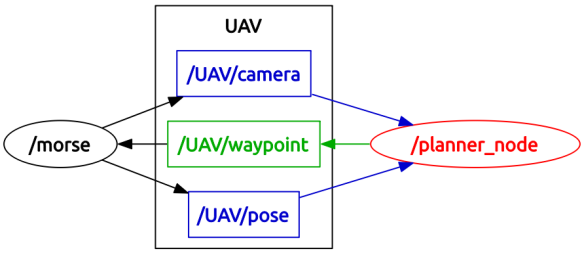

Figure 3: Graph of the ROS nodes interacting during a simulation. The communication with the sensors and the actuators happens via messages, here indicated as *camera*, *waypoint* (the destination plot), and *pose* (the current position of the UAV). Data are processed in nodes *morse* (simulation) and *planner_node* (planning).

date the MRF and to check the replanning conditions, and the cycle continues.

We considered crop fields of size corresponding to an average crop field. A field was divided into a regular grid of 425 plots. Weeds can be structured into patches and depending on the weed species, the crop and the period of the year, these patches can be more extended into tillage direction (Johnson, Mortensen, and Gotway 1996), since dispersion is made easier. Therefore, we considered two MRF models of weeds spatial distribution: an isotropic model (M1) and its anisotropic version (M2), defined by the order 2 potential functions in Eq.1 that encode spatial correlation.

$$\mathbf{M1}: \log[f_{i,j}(x_i, x_j)] = \beta \cdot \left(1 - \frac{|x_j - x_j|}{K}\right), \ \beta \in \mathbb{R}$$

$$\mathbf{M2}: \log[f_{i,j}(x_i, x_j)] = \beta_t \cdot \left(1 - \frac{|x_i - x_j|}{K}\right) \cdot \mathbf{1}_{(i,j)\in E_t} +$$
$$+ \beta_o \cdot \left(1 - \frac{|x_i - x_j|}{K}\right) \cdot \mathbf{1}_{(i,j)\in E_o}, \ (\beta_t, \beta_o) \in \mathbb{R}^2$$

where for model M2, E_t denotes the set of neighbours sites in the direction of tillage, while E_o is the set of neighbours sites in the orthogonal direction.

With these two models, assuming no prior information is available about a dominant abundance class in the map, we considered that all order 1 potential functions, $f_i(x_i)$, are equal to one. Then, the maximal order 2 weight is given when neighbouring sites i and j are in the same state, and this weight decreases when the absolute difference between x_i and x_j increases. We considered 4 abundance classes ($K = 4$) and the parameters were fixed to the values of $\beta = 2$, and $\{\beta_t, \beta_o\}$ to $\{4, 1\}$, corresponding to realistic values for weeds maps split in plots of $9m^2$. Then, using the Gibbs sampling algorithm (Koller and Friedman 2009), we generated 150 abundance maps of weeds for model M1 and for model M2. On each map we applied our online planning approach and we compared the map estimated from data sampled during the UAV trajectory with the real one. Obtaining the expected quality gain as in Eq. 4, for each possible sample in each site is a costly operation, as it implies generating a new set A of sampling sites; hence we limit these calculations in two ways. First, when updating the probability distribution after an observation, we update only those sites that have their maximum marginal probability affected by more than 1% by the observation; this corresponds to sites in a radius of 12 m from the plot currently being observed. Second, we elaborate the new expected quality gains only when the accumulated error $\epsilon_A = \sum_{A_i \in A} |x_{A^i} - x^*_{A^i}|$, where x_{A^i} and $x^*_{A^i}$ are respectively the observed and the expected values in site A^i, exceeds a given threshold. The threshold has been empirically determined as a compromise between the on-line replanning time and the overall quality of the reconstructed map, and is fixed here to $0.6 * N$, with N the total number of plots in the field. The optimisation of this threshold is still an open problem. An excessive quality threshold would result in observations more "useful", but would also trigger many replanning episodes, consuming useful flight time, and finally reducing the total number of performed observations. Considering a "dynamic" threshold would imply to integrate the remaining execution and planning time in the planning problem itself (Burns, Ruml, and Do 2013); an intermediate solution is also possible, using an architecture permitting to manage time by paralleling execution and on-board decision, as in (Chanel, Lesire, and Teichteil-Königsbuch 2014).

For each planning phase, which does not necessarily correspond to a new evaluation of the expected quality gain, we plan to visit a number $n = 21$ of sampling sites; this value corresponds to 5% of the total number of plots in the field.

We compared the performances of our replanning platform to the greedy approach of (Peyrard et al. 2013) on simulated crop fields. We extracted a reconstructed map at each 5% fraction of the total number of sites being observed, ending when half of the field was sampled. Tables 1 and 2 show the figures relative to the spatial models M1 and M2 respectively. There, *sampling* is the percentage of observed sites, ϵ_A/N is the mean difference between the estimated abundance class and the true one, the mean being computed over all 425 sites and all 150 test fields (a lower ϵ_A/N indicates a better overall quality of the map), *CES* (for Correctly Estimated Sites) is the mean value of the ratio to N of the sites estimated to the correct abundance class, *replans* is the av-

M1	Planning				Greedy			Greedy-constrained		
sampling	ϵ_A/N	*CES*	*dist (m)*	*replans*	ϵ_A/N	*CES*	*dist (m)*	ϵ_A/N	*CES*	*dist (m)*
0%	0.77	0.37	0	0	0.71	0.39	0	0.76	0.37	0
5%	0.71	0.41	540	1	0.66	0.41	3498	0.71	0.41	3648
10%	0.67	0.45	1032	2	0.62	0.45	6219	0.66	0.45	6393
15%	0.63	0.48	1539	4	0.57	0.49	8232	0.61	0.49	8283
20%	0.58	0.52	2094	5	0.52	0.53	10044	0.57	0.53	10410
25%	0.54	0.55	2625	6	0.48	0.56	11829	0.53	0.56	12552
30%	0.49	0.59	3144	7	0.44	0.60	13785	0.48	0.60	14739
35%	0.45	0.62	3699	8	0.41	0.63	15816	0.44	0.63	17211
40%	0.41	0.66	4242	9	0.37	0.66	17784	0.42	0.65	18501
45%	0.40	0.67	4653	15	0.33	0.69	19677	0.42	0.65	18516
50%	-	-	-	-	0.42	0.72	21795	0.42	0.65	18516

Table 1: Performances of the planning approach vs. greedy adaptive sampling, on 150 test fields built on the M1 model.

M2	Planning				Greedy			Greedy-constrained		
sampling	ϵ_A/N	*CES*	*dist (m)*	*replans*	ϵ_A/N	*CES*	*dist (m)*	ϵ_A/N	*CES*	*dist (m)*
0%	0.74	0.38	0	0	0.71	0.37	0	0.75	0.37	0
5%	0.72	0.41	537	1	0.70	0.42	3429	0.69	0.42	3630
10%	0.69	0.43	1026	3	0.67	0.45	6297	0.67	0.45	6297
15%	0.66	0.46	1533	4	0.64	0.48	9051	0.63	0.48	8460
20%	0.63	0.48	2091	6	0.60	0.51	10947	0.60	0.51	10551
25%	0.61	0.51	2631	7	0.56	0.54	12789	0.57	0.54	12681
30%	0.58	0.53	3159	8	0.53	0.57	14661	0.54	0.57	14895
35%	0.55	0.55	3780	10	0.49	0.60	16710	0.50	0.60	17322
40%	0.53	0.57	4440	11	0.45	0.63	18774	0.48	0.61	18561
45%	0.42	0.66	4635	15	0.42	0.66	20748	0.48	0.61	18588
50%	-	-	-	-	0.38	0.69	23031	0.48	0.61	18588

Table 2: Performances of the planning approach versus greedy adaptive sampling, on 150 test fields built on the M2 model.

erage number of replanning episodes, *dist* is the travelled distance in meters. To evaluate the actual quality of the reconstructed map, we compare it to the real abundance map. While the quality, measured in terms of ϵ_A and *CES*, is better for the greedy approach, it is even so comparable to the results obtained by the planning platform, which can still be improved by lowering the threshold for triggering the replanning episodes. But most importantly the flying time, which is measured in terms of the distance flown, is significantly smaller with the planning approach.

The sampling driven by the planner generally ends up when 40% of the plots are observed, while the greedy approach continues until stopped at 50%. We recall that the planning model we use encodes the neighbours of an observed site as not worth visiting: the goal is thus reached when an observation action has been performed in all the plots or in one of their close neighbours. The result is a much smaller flying time, for a quality of the map close to the one obtained by the greedy adaptive sampling algorithm. We then improved the greedy model to include a similar modelling of the sampling: when selecting the next-to-visit site, the close neighbours of sites already visited were ruled out, even if observing the weed density there would lower the remaining uncertainty in the estimated complete map. We show the figures for the distances and the corresponding quality in Fig.4, comparing the three approaches on model M1 only (we observed the very same trend with M2), with a numerical constraint on distance of 4500m, which roughly corresponds to 20 minutes of flight time.

This *greedy-constrained* algorithm ends up having similar quality performance on the final map, even if the quality estimated online is worse. The flown distance however is eventually smaller than the original greedy algorithm, and the sampling process stops when 40-45% of the plots have been observed, as for the planning approach, even if the flown distance is still significantly higher compared to it. Crucially, the replanning approach ends up with a much better map quality at the distance limit.

The different behaviours in terms of ratio quality/distance of the two approaches can be ascribed both to the difference in the visiting sequence, clearly less expensive when using a planner, and to the selection criterion. Minimising the uncertainty in the abundance classes in the map generates a more expensive visiting sequence, in part because the further away the sites are from previous observations, the bigger the uncertainty on their value is: this yields to paths that fly the UAV along the borders, away from the middle of the field. The opposite behaviour can happen when maximising the expected gain: observing close to the middle of the field has more chances to bring larger gains from the neighbours sites; this trend is reduced by the limits we put on sampling neighbours. Fig.5 shows an execution on a M1 field of the

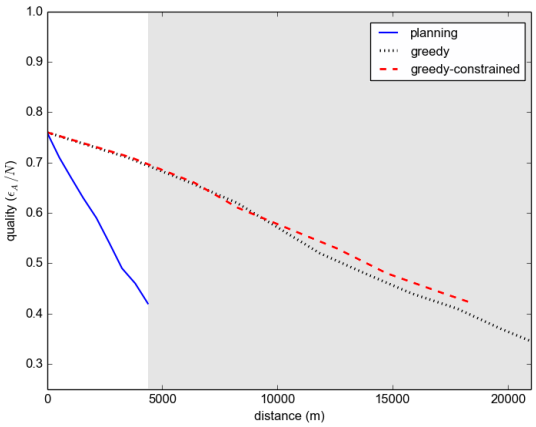

Figure 4: Plot of quality versus distance for model M1. Dotted line is the initial greedy approach, dashed line is the updated greedy approach, plain line is the replanning approach. The grey area shows the distance limit set at 4500m.

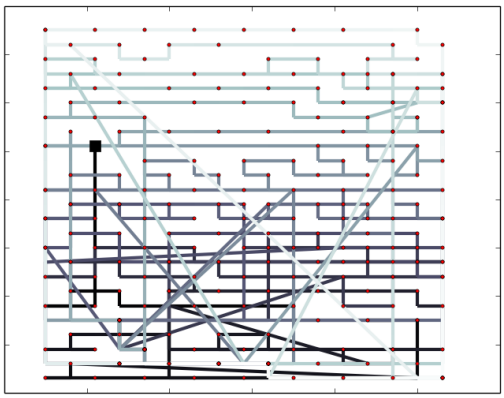

Figure 5: An execution path on the model M1. The square is the initial pose, the circles are observation sites. Clarity in the stroke indicates the advance in the execution.

planning algorithm. We can see how the UAV samples sites in small steps order, moving from the bottom to the top, and lingering in the middle. Long distances are flown when samples have covered a big part of the area, and when still non observed sites can bring some small quality gain. In comparison, the trajectory followed using the greedy method would appear quite disordered because it always moves to the next most informative site, without considering the distance and thus ignoring the potential impact on the total number of sampling sites that can be observed within the time budget. It is then obvious that such an efficient execution pattern can be certainly obtained by optimisation techniques.

Conclusions

We have described here a novel AI planning-based approach to deploy autonomous UAVs on demand without any heavy logistics, in order to sample pests in a crop field and map their spatial distribution. Currently, the most common sampling method relies on a fixed choice of sampled plots in the field, visited by humans that assess the abundance class. This solution is time consuming and in practice only a limited number of plots can be sampled. On the contrary, our approach *autonomously* produces an estimated map of pest abundance in the fields. The platform integrates Markov random fields for knowledge representation, updated at runtime by the observations of the UAV's embarked sensors.

AI planning has been recently used with success for UAV mission planning, i.e. Search-And-Rescue problems with low-cost quadcopters (Bernardini, Fox, and Long 2014) or Multi-Target Detection and Recognition with middle-size UAVs (Chanel, Teichteil-Königsbuch, and Lesire 2013). While the former application is more oriented towards target tracking, the latter is focused on dynamic data acquisition and environmental knowledge optimisation like the application we present in this paper. But while they rely on rather complex POMDP techniques, our approach assumes the use of small UAVs with limited resources, which requires light planning capabilities like our determinization-based replanning method. The same authors proposed as well a flexible replanning framework for probabilistic POMDP-like planners and robotic architectures (Chanel, Lesire, and Teichteil-Königsbuch 2014), which could be worth adapted in a near future to our deterministic fault-tolerant-like approach in order to have a finer control over planning and execution times as well as replanning strategies.

We have illustrated the planning approach to the problem of weeds mapping in crop field, on a realistic SAIL simulation platform. When compared to a greedy approach that selects the next sites to be visited by the UAV without accounting for the future flight duration, we observe that the planning approach leads to results of similar quality but at much less cost (measured as the distance covered during the flight). This means that if the same distance is allocated to the two approaches, the planner will enable to sample more plots, and therefore to provide better quality estimated maps.

Another advantage of the planning approach for UAVs, not yet exploited, is that the flight height can also be adapted in order to modify the images resolution. One could imagine starting the pest sampling with low resolution images and then focusing on areas of particular interest in the field, with higher resolution images for a better quality map estimation. We plan to represent such model with hierarchical MRFs, as a future development of the approach.

The next step is now to test the planning approach in real field conditions. In particular, the information gathered by the UAV will not be anymore directly the abundance classes in the sampled plots, but a true image of the plot. An image analysis step will be embedded into our approach to estimate the class from this image. Experiments will be conducted with an AscTec Firefly UAV for either weeds or phoma stem canker mapping in sunflower fields.

Acknowledgements This research has been funded by Région Midi-Pyrénées (grant 13/05/12.05), the LARDONS project (grant ANR-2010-BLAN-0215-04), INRA, and Onera, the French Aerospace Lab.

References

Albore, A., and Bertoli, P. 2006. Safe LTL assumption-based planning. In *Proc. of Int. Conf. on Automated Planning and Scheduling (ICAPS-06)*.

Barto, A.; Sutton, R.; and Watkins, C. 1989. Learning and sequential decision making. In Gabriel, M., and Moore, J., eds., *Learning and Computational Neuroscience*. MIT Press.

Bernardini, S.; Fox, M.; and Long, D. 2014. Planning the behaviour of low-cost quadcopters for surveillance missions. In *Proc. of Int. Conf. on Automated Planning and Scheduling (ICAPS-14)*.

Bertoli, P.; Cimatti, A.; Pistore, M.; Roveri, M.; and Traverso, P. 2001. MBP: a model based planner. In *Proc. of the IJCAI-01 Workshop on Planning under Uncertainty and Incomplete Information*.

Bonneau, M.; Gaba, S.; Peyrard, N.; and Sabbadin, R. 2014. Reinforcement learning-based design of sampling policies under cost constraints in markov random fields: Application to weed map reconstruction. *Computational Statistics and Data Analysis* 72:30–44.

Bonneau, M.; Peyrard, N.; and Sabbadin, R. 2012. A reinforcement-learning algorithm for sampling design in markov random fields. In *Proc. European Conference on Artificial Intelligence (ECAI-12)*, 181–186.

Brafman, R. I., and Shani, G. 2012. Replanning in domains with partial information and sensing actions. *J. Artif. Int. Res.* 45(1):565–600.

Burns, E.; Ruml, W.; and Do, M. B. 2013. Heuristic search when time matters. *J. Artif. Int. Res.* 47(1):697–740.

Chanel, C. P. C.; Lesire, C.; and Teichteil-Königsbuch, F. 2014. A robotic execution framework for online probabilistic (re)planning. In *Proc. of Int. Conf. on Automated Planning and Scheduling (ICAPS-14)*.

Chanel, C. P. C.; Teichteil-Königsbuch, F.; and Lesire, C. 2013. Multi-target detection and recognition by uavs using online pomdps. In *Proc. of AAAI-13*.

Cimatti, A.; Clarke, E.; Giunchiglia, F.; and Roveri, M. 2000. NuSMV: a new symbolic model checker. *International Journal on Software Tools for Technology Transfer* 2(4):410–425.

Domshlak, C. 2013. Fault tolerant planning: Complexity and compilation. In *Proc. of Int. Conf. on Automated Planning and Scheduling (ICAPS-13)*.

Echeverria, G.; Lassabe, N.; Degroote, A.; and Lemaignan, S. 2011. Modular openrobots simulation engine: Morse. In *Proceedings of the IEEE ICRA*.

Geman, S., and Geman, D. 1984. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 6:721–741.

Ivankovic, F.; Haslum, P.; Thiébaux, S.; Shivashankar, V.; and Nau, D. S. 2014. Optimal planning with global numerical state constraints. In *Proc. of Int. Conf. on Automated Planning and Scheduling (ICAPS-14)*.

Johnson, G.; Mortensen, D.; and Gotway, C. 1996. Spatial and temporal analysis of weed seedling populations using geostatistics. *Weed Science* 44(3).

Koller, D., and Friedman, N. 2009. *Probabilistic Graphical Models : Principles and Techniques*. MIT Press.

Lipovetzky, N., and Geffner, H. 2012. Width and serialization of classical planning problems. In *Proc. European Conference on Artificial Intelligence (ECAI-12)*, 540–545.

Mooij, J. M. 2010. libDAI: A free and open source C++ library for discrete approximate inference in graphical models. *Journal of Machine Learning Research* 11:2169–2173.

Murphy, K. P.; Weiss, Y.; and Jordan, M. I. 1999. Loopy belief propagation for approximate inference: An empirical study. In *Proc. of the conference on Uncertainty in Artificial Intelligence (UAI)*, 467–475. Morgan Kaufmann Publishers Inc.

Peyrard, N.; Sabbadin, R.; Spring, D.; Brook, B.; and Mac Nally, R. 2013. Model-based adaptive spatial sampling for occurrence map construction. *Statistics and Computing* 23(1):29–42.

Quigley, M.; Conley, K.; Gerkey, B.; Faust, J.; Foote, T.; Leibs, J.; Wheeler, R.; and Ng, A. Y. 2009. ROS: an open-source robot operating system. In *ICRA workshop on open source software*, volume 3.

Ramirez, M.; Lipovetzky, N.; and Muise, C. 2014. Lightweight automated planning toolkit. Technical report, http://lapkt.org/.

Roth, D. 1996. On the hardness of approximate reasoning. *Artificial Intelligence* 82:273–302.

Teichteil-Königsbuch, F.; Kuter, U.; and Infantes, G. 2010. Incremental plan aggregation for generating policies in mdps. In *9th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2010), Toronto, Canada, May 10-14, 2010, Volume 1-3*, 1231–1238.

Yoon, S. W.; Ruml, W.; Benton, J.; and Do, M. B. 2010. Improving determinization in hindsight for on-line probabilistic planning. In *Proc. of Int. Conf. on Automated Planning and Scheduling (ICAPS-10)*, 209–217.

Yoon, S. W.; Fern, A.; and Givan, R. 2007. FF-Replan: A baseline for probabilistic planning. In *Proc. of Int. Conf. on Automated Planning and Scheduling (ICAPS-07)*, 352–359.