Reinforcement learning based design of sampling policies under cost constraints in Markov Random Fields

Régis Sabbadin, Nathalie Dubois Peyrard Peyrard, Mathieu Bonneau Bonneau

REINFORCEMENT LEARNING BASED DESIGN OF SAMPLING POLICIES UNDER COST CONSTRAINTS IN MARKOV RANDOM FIELDS

BY

MATHIEU BONNEAU
NATHALIE PEYRARD
RÉGIS SABBADIN

RAPPORT DE RECHERCHE 2013-MIAT-RR-1
JUILLET 2013

Unité de Mathématiques et Informatique Appliquée du centre de Toulouse
Institut National de la Recherche Agronomique
Toulouse, France
http://carlit.toulouse.inra.fr/wikiz/index.php/Accueil

Reinforcement learning based design of sampling policies under cost constraints in Markov Random Fields

Mathieu Bonneau[a], Nathalie Peyrard[a] and Régis Sabbadin[a]

Affiliation:[1]
[a] INRA - Applied Mathematics and Computer Science Unit
24 Chemin de Borde Rouge - Auzeville
CS 52627
31326 Castanet Tolosan cedex - FRANCE
{bonneau, peyrard, sabbadin}@toulouse.inra.fr

Abstract:

Markov random fields (MRF) offer a powerful representation for reasoning on large sets of random variables in interaction. A classical, but difficult inference task is the evaluation of the most probable assignment of a variable given the values of some others (Maximum Posterior Marginal probability computation, MPM). Linked to that problem, optimising the choice of the variables to observe (a sample) in order to maximise the MPM probabilities is even more difficult. In the field of spatial statistics, the design of sampling policies has been largely studied in the case of continuous variables, using tools from the geostatistics domain. In the MRF case with discrete-valued variables, some heuristics have been proposed for the design problem but there exists no universally accepted solution, in particular when considering adaptive policies, as opposed to static ones. In this paper we formalise the problem of optimal adaptive sampling in a MRF as a finite-horizon Markov Decision Process (MDP) with a factored state space. A policy of this MDP is a non stationnary decision rule which associates a set of sampling locations to the set of past observations. Solving this MDP amounts to computing the optimal adaptive sampling policy according to a given quality criterion. The translation of the initial optimization problem into the MDP framework enables to exploit the Reinforcement Learning (RL) paradigm and to propose an original al-

[1]**Corresponding author:** Nathalie Peyrard
INRA - Applied Mathematics and Computer Science Unit
24 Chemin de Borde Rouge - Auzeville
CS 52627
31326 Castanet Tolosan cedex - FRANCE
Email: peyrard@toulouse.inra.fr
Tel : +33(0)5.61.28.54.39
Fax : +33(0)5.61.28.53.35

gorithm for its approximate resolution. This generic procedure, named Least Square Dynamic Programming (LSDP), combines a parameterized representation of the value of a policy, the construction of a batch of simulated trajectories of the MDP and a backwards induction algorithm. It is not only dedicated to the optimal adaptive sampling problem but can be used to solve any factored MDP under finite horizon. Then LSDP can be specialized to solve the above-mentioned sampling problem. Based on an empirical comparison of the performance of LSDP with existing one-step-look-ahead sampling heuristics and solutions provided by classical RL algorithms, the following conclusions can be derived: (i) a naïve heuristic, consisting in sampling sites where marginals are the most uncertain, is already an efficient sampling approach. (ii) LSDP outperforms all the classical RL approaches we have tested. (iii) LSDP outperforms the heuristic approach in cases when sampling costs are not uniform over the set of variables, or sampling actions are constrained.

Keywords: Heuristic and optimal sampling design, sampling cost, dynamic programming, Markov decision process, weed mapping

1. INTRODUCTION

The questions of building probabilistic models of spatial processes and building plausible reconstructions of the process from the model and observed data are classic and have mobilized several research fields in spatial statistics or probabilistic graphical models communities. Nearly as classical is the question of designing optimal *sampling policies* allowing to build reconstructions of high probability when the model is known. This question is more complex to solve than the pure reconstruction problem and cannot be solved optimally in general. This sampling design problem has been tackled in spatial statistics [7, 20] and artificial intelligence [15, 16, 25]. It is even more complex in the case of adaptive sampling, where the set of sampled sites is chosen sequentially and observations from previous sampling steps are taken into account to select the next sites to explore [33].

The case of sampling real-valued observations (e.g. temperature or pollution monitoring) has been the most studied, mainly within the geostatistical framework of Gaussian random fields and kriging. Much less attention has been paid to the case of sampling variables with finite state space. However, this problem arises naturally in many studies about biological systems, where observations can be species abundance classes, disease severity classes, presence/absence values. In this article, we focus on this case and propose, similarly to [15, 25, 26], to define the corresponding optimal sampling problem within the framework of Markov random fields (MRF, [10]). MRF are well adapted to model variables with finite state space. They are, for instance, very popular in image analysis to model image segmentation problems. A sampling policy can be *static* or *adaptive*. In the first case, the set of sampled sites is chosen once and for all at the beginning of the survey (see [9] for a recent work on static sampling of counts data). With an adaptive policy, the survey is divided into successive steps and the next set of sampled sites is chosen according to previous observations. Obviously, adaptive policies are more efficient than static ones, but may not always be applicable. In [15], the authors considered the sampling problem in a particular case of MRF, defined on

polytrees. They looked for static sampling policies, as in [25]. The work in [26] was the first proposition of a naive heuristic solution to design an adaptive sampling policy for the general MRF model. The heuristic was derived from a strong simplification of the model. Here we extend the work of [26] by proposing a heuristic policy built from simulations of the exact MRF model. For this, we propose to encode the optimal adaptive sampling problem as a finite-horizon Markov Decision Process (MDP, [28]) with factored state space. A policy for this MDP is a set of non stationary decision rules (one per sampling step) which associate a set of sampling locations to the set of past observations. Thus the MDP optimal solution provides an optimal adaptive sampling policy for the MRF reconstruction problem, according to a given quality criterion. Solving MDPs with factored state space is hard. It is no easier than solving the adaptive sampling problem in MRF. However, casting the optimal sampling problem within the MDP framework allows us to exploit principles from the family of *Reinforcement Learning* (RL, [31]) approaches which have been proposed to solve approximately large (or factored) state space MDPs.

RL approaches allow to solve MDPs approximately by making use of simulations of the process dynamics. They can be used *on-line* to construct adaptive policies step-by-step, computing only the current action to apply from the set of past observations, or they can be used *off-line*, computing a complete policy before any observation is actually made. Off-line approaches focus their computational effort prior to policy execution, while on-line approaches alternate action computation phases and action execution phases. The approach we propose in this paper is an off-line RL algorithm.

As we will demonstrate, classical RL algorithms cannot be applied to solve the optimal sampling problem without being adapted. Therefore we provide a new generic RL algorithm that can be used to solve approximately any large state-space finite horizon MDPs: the *Least Square Dynamic Programming* algorithm (LSDP). LSDP relies on three main ingredients: (1) the value function of a policy is parameterized as a linear combination of features; (2) simulated trajectories of the MDP are computed off-line and stored in a *batch*; (3) the weights of the linear approximation are those which minimize the least-square error evaluated on the simulated trajectories. We then show how to specialize this generic algorithm to the problem of optimal adaptive sampling in MRFs. We show experimentally that this algorithm improves over classical "one-step-look-ahead" heuristics and RL approaches, thus providing a reference algorithm for sampling design.

This paper starts with a description of the case study that motivated this work: weed sampling in a crop field (Section 2). Then, the MRF formalization of the optimal adaptive spatial sampling problem is introduced in Section 3. We show how to model it as a finite-horizon factored MDP in Section 4 and we discuss classical RL solutions for computing approximations of the optimal policy, in Section 5. Then, we describe the LSDP algorithm in Section 6 and its application to the problem of sampling in MRF in Section 7. We present an empirical comparison between one-step-look-ahead approaches, classical RL algorithms and LSDP, on toy problems and on the weed sampling problem in Section 8. Some methodological and applied perspectives of this work are discussed in Section 9.

2. CASE STUDY: WEED SAMPLING IN A CROP FIELD

In arable fields, weeds are responsible for yield loss [22] because they are competing with crop for resources and they can be host for parasites and diseases. In the meantime, the role of weeds in agro-ecosystem food webs and in providing ecological services has been established [11]. Therefore new weeds management policies have to be designed. As a consequence, as a prerequisite to management, there has been a growing interest for the study of the spatial repartition of weeds in crops. Such studies are usually based on maps.

A map of weed density classes is usually built from a probabilistic model of weed spatial repartition and from a sample output. The field is decomposed into a set of quadrats and a sample is a subset of the quadrats. The sample output is the set of density classes assessed at each sampled quadrat. Observation of weeds and class assessment is very time consuming. It requires people with profound knowledge in order to recognize the different weed species. Therefore a crucial problem is to choose which quadrats in the field should be observed in order to ensure a reconstructed map of good quality for a reasonable sampling time. A lot of attention has been paid to the development of spatial sampling methods for weed mapping [6, 34]. However, none are adaptive. In this paper we will propose a method to design adaptive policies of weed sampling. When applying an adaptive policy, the sampling procedure is divided into successive steps and the quadrats sampled at a given step are given by a decision rule which associate previous observations (locations and density classes) to a set of quadrats. To apply our method and compute the corresponding adaptive sampling policy we will provide a model of the joint distribution of the density classes at each quadrat as that of a Markov random field (MRF) and a model of sampling cost based on the time spent for assessing the density class into a quadrat (see Section 8).

3. OPTIMAL ADAPTIVE SAMPLING IN MARKOV RANDOM FIELDS

In this section we formalize the problem of optimal adaptive sampling in a MRF.

3.1. Problem statement

Let $X = (X_1, \ldots, X_n)$ be a vector of discrete random variables taking values in $\Omega^n = \{0, \ldots, K\}^n$. $V = \{1, \ldots, n\}$ is the set of indices of the vector X and an element $i \in V$ will be called a *site*. The distribution $\mathbb{P}$ of X is that of a MRF (also named *undirected graphical model*) with associated graph $G = (V, E)$, where $E \subseteq V^2$ is a set of undirected edges (see Figure 1 for an example). The vector $x = (x_1, \ldots, x_n)$ is a realization of X and we adopt the following notation: $x_B = \{x_i\}_{i \in B}$, $\forall B \subseteq V$. Then the joint distribution of X is a Gibbs distribution: $\mathbb{P}(X = x) \propto \prod_{c \in \mathcal{C}} \Psi_c(x_c)$, where $\mathcal{C}$ is the set of cliques of V and the $\Psi_c, c \in \mathcal{C}$, are strictly positive potential functions [14].

Let us assume that we want to reconstruct the vector X on a specified subset $R \subseteq V$ of sites of interest and that, for this, we can only acquire a limited number of observations on a subset $O \subseteq V$ of observable sites. We will assume that $R \cup O = V$ and intersection between O and R can be non-empty. The sampling problem is then to select a set of sites $A \subseteq O$, named a *sample*, where X will be observed. When

sample A is chosen, a *sample output* x_A results, from which the MRF distribution $\mathbb{P}$ is updated. Intuitively, our objective is to choose A so that the updated distribution $\mathbb{P}(\cdot|x_A)$ becomes as *informative* as possible, in expectation over all possible sample outputs x_A. We measure how informative a distribution is by using some classical information criteria such as *Maximum Posterior Marginals* or *Maximum A Posteriori* (see next paragraph).

In the following we describe the different elements allowing to formally define the problem of Optimal Adaptive Sampling in a MRF (OASMRF).

Reconstruction. When a sample output x_A is available, the *Maximum Posterior Marginal* (MPM) criterion can be used to derive an estimator x_R^* of the hidden map x_R:

$$x_R^* = \left\{ x_i^* \quad | \quad i \in R, \quad x_i^* = \operatorname*{argmax}_{x_i \in \Omega} \mathbb{P}(x_i \mid x_A) \right\}.$$

Alternately, other reconstruction criteria, such as the *Maximum A Posteriori* (MAP) criterion [25] could be considered. The hidden map would be then reconstructed as the mode of the joint conditional distribution $\mathbb{P}(x \mid x_A)$.

Adaptive sampling policy. In adaptive sampling, the sample A is chosen sequentially. The *sampling plan* (the sequence of samples) is divided into H steps. $A^h \subseteq O$ is the sample explored at step $h \in \{1, \ldots, H\}$ and x_{A^h} is the sample output at step h. The sample size is bounded ($|A^h| \leq L$) and Δ_L is the set of all policies satisfying $|A^h| \leq L, \forall h$. The choice of sample A^h depends on the previous samples and outputs. An adaptive sampling policy $\delta = (\delta^1, \ldots, \delta^H)$ is defined by an initial sample A^1 and functions (decision rules) δ^h specifying the sample chosen at step $h \geq 2$, depending on the results of the previous steps: $\delta^h((A^1, x_{A^1}), \ldots, (A^{h-1}, x_{A^{h-1}})) = A^h$. Therefore the policy describes the decision rules for all possible hidden maps x. A trajectory $(A^1, x_{A^1}), \ldots, (A^H, x_{A^H})$ is the succession of samples and sample outputs encountered when applying policy δ to a particular realization x. The set of all trajectories which can be followed by policy δ is τ_δ. We will assume throughout the paper that observations are reliable. As a consequence, we will only consider policies visiting each site at most once ($A^h \cap A^{h'} = \emptyset, \forall h \neq h'$).

Figure 1 shows an example of an adaptive sampling policy, in the case where $H = 3$, $L = 1$ and $K = 1$. A static policy is a particular case where the sample visited at each step is the same for each trajectory.

Quality of a sampling policy. The quality of a policy δ is measured as the expected quality of the estimator x_R^* that can be obtained from δ. In practice, since applying a policy δ may lead to different trajectories $((A^h, x_{A^h}))_{h=1..H}$, each trajectory having its own probability to occur, it is useful to first define the quality U of a trajectory $((A^h, x_{A^h}))_{h=1..H}$ as a function of $(\bar{A}^H, x_{\bar{A}^H})$, where $\bar{A}^H = \cup_{h=1..H} A^h$:

$$U\big(((A^h, x_{A^h}))_{h=1..H}\big) = U\big(\bar{A}^H, x_{\bar{A}^H}\big) = \sum_{i \in R} \max_{x_i \in \Omega} \Big\{ \mathbb{P}(x_i \mid x_{\bar{A}^H}) \Big\}. \qquad (1)$$

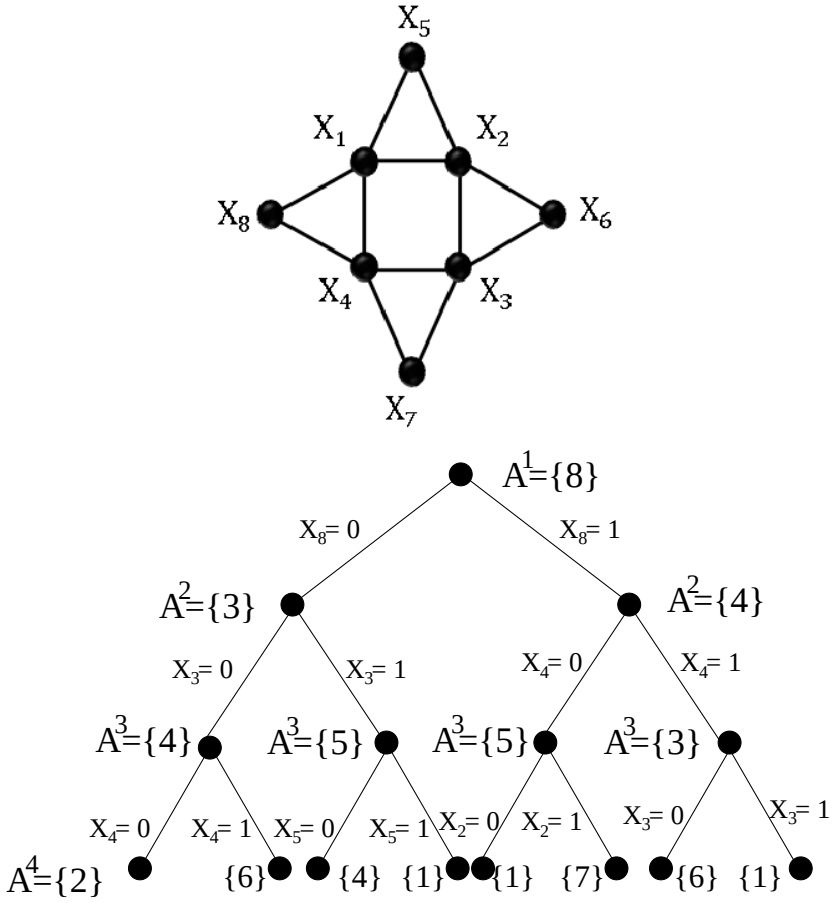

Figure 1: Tree representation of an adaptive sampling policy. Top: graph G associated to the MRF model ($n = 8$). Bottom: policy tree representation of an adaptive policy for sampling in this MRF, for $L = 1$, $H = 3$ and $K = 1$ (the final sample outputs are not represented). Two different realisations x of the MRF will lead to two different trajectories in the policy tree.

The quality of a trajectory is directly related to our mapping purpose since it is equal to the expected number of well reconstructed sites when the spatial repartition model is $\mathbb{P}$ and the only available information is $x_{\bar{A}^H}$. The quality of a sampling policy δ is then defined as an expectation over all possible trajectories:

$$V(\delta) = \sum_{(\bar{A}^H, x_{\bar{A}^H}) \in \tau_\delta} \mathbb{P}\big(x_{\bar{A}^H}\big) U\big(\bar{A}^H, x_{\bar{A}^H}\big).$$

Remark that this definition of the quality of a sampling policy can be adapted to the MAP criterion. It amounts to replacing the sum of the local conditional mode probabilities by the joint conditional mode probability. In section 7 we will show that our approach can also be adapted to an *entropy* definition of trajectory quality.

Optimal adaptive sampling in MRF (OASMRF). Finally the problem of optimal adaptive sampling amounts to finding the policy of highest quality:

$$\delta^* = \arg\max_{\delta \in \Delta_L} V(\delta). \tag{2}$$

Note that since our definition of the quality of a trajectory is based on the MPM criterion, it does not depend on the order in which observations are received (this is also true for MAP and entropy-based criteria). Therefore, the optimal policy δ^* has the property that the choice of the next sample at step h depends only on $(\bar{A}^{h-1}, x_{\bar{A}^{h-1}})$:

$$\delta^{*h}\Big((A^1, x_{A^1}), \ldots, (A^{h-1}, x_{A^{h-1}})\Big) = \delta^{*h}\Big((\bar{A}^{h-1}, x_{\bar{A}^{h-1}})\Big).$$

This will be rigorously demonstrated in Proposition 1 (Section 4). The fact that the optimal action choice only depends on the union of past observations does not imply that the above defined sampling problem is not adaptive. Indeed, sample choices and observations are interleaved, meaning that the optimal policy δ^* is a function. In particular, it is true that, given a fixed set of observations, the order in which they were received influences neither the probability of this set, nor the quality of the map that can be constructed from it. However, new samples are chosen with respect to the set of observations obtained so far, which makes the OASMRF problem, both sequential (a sequence of samples choice) and adaptive (two different hidden maps will lead to two different sequences of samples, for a fixed policy). On the contrary, in a static sampling problem, the optimal policy would be independent of the hidden map x.

Example 1. *Weed sampling in a crop field.*
For our case study, quite naturally the modeling choices are the following: n is the number of quadrats, the edges in the graph G link first order neighbor quadrats, X_i is the weed density class in quadrat i, and Ω is the set of possible classes. For a given adaptive sampling policy and a given hidden map x, a trajectory is the succession of pairs of sampled quadrats and corresponding observed density classes at every sampling steps. In practice we will consider only policies that explore one quadrat per step ($L = 1$).

Exact resolution of the OASMRF problem is intractable (see [5] for a complexity study). In the next section we will present a factored MDP model of this optimization problem. It will allow us to solve it approximately by applying Reinforcement Learning (RL) principles [31].

3.2. How to represent and handle cost constraints?

So far, we have considered a sampling budget in terms of a number, H, of allowed sampling steps and a fixed number L of sampled variables per step. This has been defined regardless of any notion of observation costs. In this section, we will discuss more general sampling cost models and discuss how they can be handled within the proposed OASMRF model.

Optimizing a trade-off between restoration quality and cost. In order to optimize a global trade-off between restoration quality and cost it is possible to include a measure of sampling cost, which has to give values commensurate with the restoration quality. Then, cost and quality measures can be added in the definition of U, to form the adaptive sampling policy quality.
However, one may question the assumption that cost and restoration quality measures be commensurate. One way to avoid this assumption is to consider a sample budget constraint instead of including sample costs into the function U, as we suggest next.

Maximizing restoration quality under cost constraint. This choice will be the one used by our procedure. Instead of considering that $L \times H$ sites can be sampled, we can consider a global (integer-valued) sampling budget B (e.g. a time budget), and assume that each subset $A \in O$ has a different (integer) sampling cost, denoted $S_C(A)$. It may as well be that sample costs depend on the observed values of the variables, $x_A : S_C(A, x_A)$. This cost function S_C is not used in the definition of U, but rather for the definition of the space of allowed trajectories, $\{(A^1, x_{A^1}), (A^2, x_{A^2}) \ldots\}$. In that case the length of a trajectory is no more constant. It depends on the number of samples needed to exhaust the budget and of the corresponding observations. To make understanding easier and to simplify notations, we will consider throughout the paper that trajectories have constant length ($S_C(A, x_A) = 1$). But we will see in Sections 6 and 8 that the LSDP procedure proposed in this paper to solve approximately (2) can handle the more general case where $S_C(A, x_A)$ is not constant.

Minimizing cost under restoration quality constraint. Conversely, one could also consider that the objective is to minimize the sampling budget, and that we are given a MPM restoration quality threshold. Sampling should be continued until the restoration quality threshold is met, and then stopped. Then, the optimization problem would be to find the sampling policy of minimum expected cost, which allows to compute restored maps which quality is above the fixed threshold. One difficulty to apply the simulation-based procedure proposed in this paper would be that the MPM value has to be computed at every sampling step, in order to check whether the end of a trajectory is reached. These computations would make the optimization problem harder to solve.

Example 2. *Weed sampling in a crop field.*
For weed sampling, the cost of a trajectory is defined as the sum of the cost of each sampling step. The cost for a given step is defined as $S_C(i, x_i)$, a function of both the sampled quadrat i and the corresponding observed density class, x_i. It represents the time needed to assess the weed class in a quadrat and depends both on the class of the weed on interest and on the number and classes or other weeds present in quadrat i.

4. FINITE HORIZON MDP MODELLING OF THE OASMRF PROBLEM

A finite-horizon MDP model [28] is a 5-tuple $< \mathcal{S}, \mathcal{D}, T, p, r >$, where $\mathcal{S}$ is a finite set of system *states*, $\mathcal{D}$ is a finite set of available *decisions*, $T = \{1, \ldots, H\}$ is a finite set of decision steps, termed *horizon*. p is a set of *transition functions* $p^t, t = 1 \ldots H$, where $p^t(s^{t+1}|s^t, d^t)$ indicates the probability that state $s^{t+1} \in \mathcal{S}$ results when the system is in state $s^t \in S$ and decision $d^t \in \mathcal{D}$ is implemented at time $t \in T$. A *terminal state* $s^{H+1} \in \mathcal{S}$ results when the last decision is applied, at decision step H. r is a set of *reward functions*: $r^t(s^t, d^t) \in \mathbb{R}$ is obtained when the system is in state s^t at time t and d^t is applied. A *terminal reward* $r^{H+1}(s^{H+1})$ is obtained when state s^{H+1} is reached at time $H + 1$.

A *decision policy* (or *policy*, for short), $\pi = \{\pi^1, \ldots, \pi^H\}$, is a set of decision functions $\pi^t : \mathcal{S} \to \mathcal{D}$. Once a decision policy is fixed, the MDP dynamics becomes that of a finite Markov chain over $\mathcal{S}$, with transition probability $p^t(s^{t+1}|s^t, \pi^t(s^t))$. The *value function* $V^\pi : \mathcal{S} \times T \to \mathbb{R}$ of a policy π is defined as the expectation of the sum of future rewards, obtained from the current state and time step when following the Markov chain defined by π:

$$V^\pi(s, t) = \mathbb{E}_\pi \left[\sum_{t'=t}^{H} r^{t'}(S^{t'}, \pi(S^{t'})) + r^{H+1}(S^{H+1}) \mid S^t = s \right], \forall (s, t) \in \mathcal{S} \times T.$$

Here the notation S^t in capital letter represents the random variable associated to the state at time t, while s^t is a possible realization.
Solving a MDP amounts to finding an *optimal policy* π^* which value is maximal for all states and decision steps: $V^{\pi^*}(s, t) \geq V^\pi(s, t), \forall \pi, s, t$ (it can be proved that there always exists at least one optimal policy, see [28]).

We now model the OASMRF problem in the MDP framework. It corresponds to the graphical representation of Figure 2.

State space. state $s^t, t = 1, \ldots, H+1$ summarizes current information about variables indexed in O: if sample $\bar{A}^{t-1}$ was explored and observations $x_{\bar{A}^{t-1}}$ were obtained at previous sampling steps,

$$s^t = (\bar{A}^{t-1}, x_{\bar{A}^{t-1}}), \forall t = 2, \ldots, H + 1 \text{ and } s^1 = (\emptyset, \emptyset).$$

It may be convenient to model s^t as a vector of length $|O|$, where $s_i^t = -1$ if site i has not yet been sampled, and $s_i^t = k$, $k \in \Omega$ if value k has been observed on site i. For instance, let us consider the case where $O = R = V = \{1, 2, 3, 4\}$ and $K = 2$. If at time $t = 1$ site 1 is observed in state 1 and at time $t = 2$ site 3 is observed in state 1,

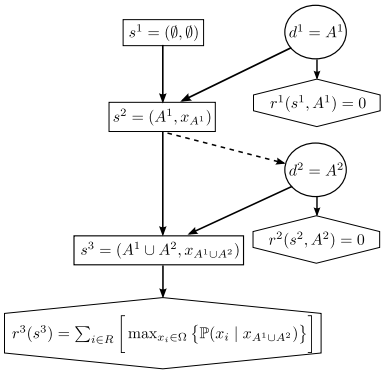

Figure 2: MDP model of a OASMRF problem with horizon $H = 2$. Rectangular nodes represent state values, circles represent actions and diamonds represent immediate rewards.

then we have the two notaitons $s^3 = (1, -1, 1, -1)$ or $s^3 = ((1, 3), (1, 1))$.

Decision space. A decision d^t is a sample $A^t \subseteq O$ such that $|A^t| \leq L$. In practice, it will never be optimal to sample a site twice (since observations are assumed to be reliable). So, we can restrict the set of decisions to those satisfying $d^t \cap d^{t'} = \emptyset, \forall t' < t$. Continuing of the above example, if $d^3 = 2$ and site 2 is observed in state 2, we have $s^4 = (1, 2, 1, -1)$ or $s^4 = ((1, 2, 3), (1, 2, 1))$. Note that if s^t and s^{t+1} are given, it is straightforward to recover the decision d^t.

Horizon. Decision steps in the MDP correspond to decision steps in the OASMRF problem. Thus, $T = \{1, \ldots, H\}$.

Transition functions. If $s^t = (\bar{A}^{t-1}, x_{\bar{A}^{t-1}})$ and $d^t = A^t$, the transition function of the MDP can be derived straightforwardly from the original MRF distribution $\mathbb{P}$:

$$p^t\big(s^{t+1} \mid s^t, d^t\big) = \mathbb{P}\big(x_{A^t} \mid x_{\bar{A}^{t-1}}\big), \forall t \in T,$$

where x_{A^t} is the realization of X_{A^t} encoded in s^{t+1}. Note that for all states s^{t+1} corresponding to observations incompatible with state s^t, this transition probability will be zero.

Reward functions. $\forall t = 1, \ldots, H$, rewards are set to zero:

$$r^t(s^t, d^t) = 0, \quad \forall t \in T, s^t, d^t.$$

Note that rewards could be non zero if the objective was to optimize a trade-off between restoration quality and sampling costs (see Section 3.2).

After decision d^H has been applied at decision step H, and state $s^{H+1} = (\bar{A}^H, x_{\bar{A}^H})$ has been reached, a final reward $r^{H+1}(s^{H+1})$ is obtained. It is defined as the quality of the MPM reconstruction (see equation (1)):

$$r^{H+1}(s^{H+1}) = \sum_{i \in R} \left[\max_{x_i \in \Omega} \left\{ \mathbb{P}(x_i \mid x_{\bar{A}^H}) \right\} \right].$$

The optimal policy for the above-defined MDP is a set of functions associating new samples to unions of past sample outputs. It thus has the same structure as an OASMRF adaptive sampling policy. Furthermore, we can establish the following proposition:

Proposition 1. *An optimal policy for the MDP model of an OASMRF problem provides an optimal adaptive policy for the initial OASMRF problem.*

Proof (Sketched). The proof is only sketched here, the full version is in the Appendix section. The proof follows three steps and uses the fact that the quality of a policy does not depend on the order in which observations are obtained:

(i) We define a function ϕ, transforming any MDP policy π into a valid OASMRF policy $\delta = \phi(\pi)$, which defines decisions independently of the order in which past observations were received, and show that $V(\phi(\pi)) = V^{\pi}((\emptyset, \emptyset), 1)$.

(ii) We establish that, for any partial history (past observations), the value of an optimal OASMRF policy starting from these observations does not depend on the order in which they were received. As a consequence, we can limit the search for optimal policies of the OASMRF problem to policies prescribing decisions which do not depend on the order of observations.

(iii) We show that any such OASMRF policy δ can be transformed into a MDP policy, through a transformation μ, and that $V(\delta) = V^{\mu(\delta)}((\emptyset, \emptyset), 1)$.

As a result of these three steps, if π^* is an optimal policy for the MDP encoding of the OASMRF problem, then $\phi(\pi^*)$ is optimal for the OASMRF problem.

□

In the following we will use the same notation δ to represent both OASMRF and MDP policies.

The finite-horizon MDP model of the OASMRF problem has state and action spaces of exponential size in the size of the original problem. However, explicit representation of the problem (and its solution policy) can be avoided, thanks to the use of RL algorithms. We describe the RL approach in the next section.

5. CANDIDATE APPROACHES FOR SOLVING OASMRF

5.1. Exact dynamic programming

Let us define the state-action value function, also called Q-function associated to any policy δ of a finite-horizon MDP problem:

$$Q^{\delta}(s, d, t) = r^t(s, d) + \sum_{s' \in Succ^t(s,d)} p^t(s'|s, d) V^{\delta}(s', t+1).$$

This function represents the expected reward when applying decision d in state s at time t and thereafter following policy δ. $Succ^t(s, d) = \{s', p^t(s'|s, d) > 0\}$ is the set of possible successors of s when d is applied at time t. In the OASMRF problem the

size of $Succ^t(s,d)$ can be small. From the Q-function of a policy δ, it is straightforward to compute δ and its value V^δ. The *backwards induction* algorithm [28] is based on this property and computes exactly the optimal policy δ^* of any finite-horizon MDP. It consists in, first, initializing the value function of the optimal policy at time $H+1$ for all possible final states:

$$V^*(s, H+1) = r^{H+1}(s),$$

and then solving iteratively the following equations:

$$\begin{array}{rcll} \forall t = H, \ldots, 1 & \text{and} & \forall s, d \in \mathcal{S} \times \mathcal{D}, & \\ Q^*(s,d,t) & = & r^t(s,d) + \displaystyle\sum_{s' \in Succ^t(s,d)} p^t(s'|s,d) V^*(s', t+1), & \\ V^*(s,t) & = & \max_d Q^*(s,d,t). & \end{array}$$

At each iteration the optimal policy is progressively built, as follows:

$$\delta^*(s,t) = \arg\max_d Q^*(s,d,t).$$

Here V^* and Q^* are simpler notations respectively for V^{δ^*} and Q^{δ^*}. Remark that the last equation can be written for any policy δ and not only the optimal one. Therefore any policy δ is totally specified by its Q-function, Q^δ. However, since the OASMRF problem typically involves a factored state space, $|\mathcal{S}|$ is huge, so the above system has too many equations and variables ($|\mathcal{S}| \times |\mathcal{D}| \times H$) to be used. Therefore, we have to look for approximate solution methods instead of exact ones. To do this, we can explore two families of heuristic approaches for solving OASMRF: *one-step-look-ahead approaches* and *reinforcement learning based* approaches.

5.2. Heuristic approaches

Heuristic approaches are methods for sample selection which provide an arbitrary (most likely suboptimal) sample in reasonable time. These methods can either (i) solve exactly a simpler optimization problem which approximates the original one or (ii) provide policies maximizing a function which approximates the optimal Q-function, Q^*.

One-step-look-ahead heuristics. One-step-look-ahead heuristics provide policies that optimize (exactly or approximatively) the immediate reward. This amounts to adopt a simpler definition of the Q-function, where $V^\delta(s', t+1)$ is simply replaced by $r^{t+1}(s^{t+1}, \delta(s^{t+1}))$. Involved rewards can also be approximated.

Such heuristics have been proposed, either in Statistics or in Artificial Intelligence, which can be applied to solve the OASMRF problem. In spatial sampling of natural resources, random and regular sampling are classic heuristic approaches [7]. Another classical method to sample 0/1 variables is Adaptive Cluster Sampling (ACS, [33]). Recently, [26] proposed a heuristic (*BP-max heuristic*), which consists in sampling, at each decision step, locations where the conditional marginal probabilities are the least informative (i.e. the closest to $\frac{1}{2}$ in the 0/1 case), in order to solve (2), (see Section 7.1 for a formal definition). It has been shown, experimentally, to outperform random,

Heuristic	Description	Properties
BP-max	at each step, sampled sites are those with highest remaining uncertainty in their maximal marginal	coarse approximation of the original MRF, but fast
MI	optimization of the mutual information	intractable for MRF with more than a few nodes (20)
$TD(\lambda)$	estimation of the optimal Q-function by reinforcement learning combined with tabular representation	intractable for MRF with more than a few nodes (20)
LSPI	estimation of the optimal Q-function by reinforcement learning combined with linear approximation	not adapted to finite-horizon MDP

Table 1: Summary of existing candidate heuristic methods to solve the OASMRF problem. Note that BP-max is fast, while the other methods are very time consuming.

regular and ACS heuristics. In [16], the authors proposed to optimize a mutual information (MI) criterion to design sampling policies in Gaussian fields. However, we will see that exact application of this method is out of reach for problems with more than a few nodes (see Section 8). Finally, generalization to n-steps-look-ahead heuristics is presented in [19].

Reinforcement learning based approaches. The main idea of RL approaches ([31]) is to use repeated simulated *experiences* (s^t, d^t, r^t, s^{t+1}), instead of exact dynamic programming, in order to estimate Q^*. As opposed to one-step-look-ahead heuristics, the exact definition of the Q-function is used but $V^\delta(s', t+1)$ is now approximated by an average over simulations. In practice, the approximation of Q^* can be computed for each triple (s, d, t), for example using the TD(λ) algorithm [31]. This algorithm is known to asymptotically converge to the Q^* function but cannot be applied to large factored MDPs, since the number of triplets (s, d, t) becomes huge. Then a common method is to compute a linear approximation of the Q-function: $Q^w(s, d, t) =< w, \phi(s, d, t) >$, where $w \in \mathbb{R}^m$ is a vector of parameters values and the vector of features $\phi : (\mathcal{S}, \mathcal{D}, T) \to \mathbb{R}^m$ is a mapping from state-action pairs to real-valued m-dimensional vectors. Simulations are used to compute values of w that lead to a good approximation of Q^*. In general little can be said about the convergence of such algorithms and no universal properties are known. However, in some cases, performance bounds [2, 21] or convergence guarantees [18, 32] can be found. Algorithms for computing w for a specific choice of features are, for example, *LSPI* [17] and *Fitted Q-iteration* [8, 23]. A review of approximate methods for solving MDP can be found in [3] or in [27].

6. LEAST SQUARES DYNAMIC PROGRAMMING

We now present the procedure we propose to compute an approximate solution to the original OASMRF problem. This procedure, referred to as the Least Square

Dynamic Programming (LSDP) algorithm, is not limited to this problem. It can be used to solve any finite-horizon MDP as long as the model (transitions and rewards) is known explicitly. So we first describe it in its general form and then show how it can be specialized to solve the OASMRF problem.

6.1. Main principles of LSDP

To approximate Q^*, the main ideas of the algorithm we propose is to combine i) a time-dependent parameterized representation of the Q-function, ii) least-squares estimation of the parameters, iii) Dynamic Programming iterations and iv) a sampling of the complete least-squares system built from simulations of MDP transitions.

Parameterized representation of the Q-function. As in LSPI or fitted Q-iteration, we consider an approximation Q^w of Q^* as a linear combination of m arbitrary *features*. When the horizon is finite, the optimal policy needs not be stationary. Therefore, we propose to define a different set of weights $\{w_i^t\}_i$ for each time step t:

$$\begin{aligned} Q^*(s,d,t) \approx Q^w(s,d,t) &= \sum_{i=1..m} w_i^t \phi_i(s,d,t), \forall s \in \mathcal{S}, d \in \mathcal{D}, t \in T, \text{ and} \\ Q^*(s,H+1) = Q^w(s,H+1) &= r^{H+1}(s), \forall s \in \mathcal{S}. \end{aligned}$$

Least-squares estimators. For a given step t the weights $\{w_i^t\}_i$ are computed using the approximate version of the dynamic programming equations. They are computed as the least-squares estimators associated to the following system:

$$\begin{aligned} &\forall (s,d) \in \mathcal{S} \times \mathcal{D}, \\ &\sum_{i=1..m} w_i^t \phi_i(s,d,t) = r^t(s,d) + \sum_{s' \in Succ^t(s,d)} p^t(s'|s,d) V^w(s',t+1), \\ &\text{where } V^w(s,t+1) = \max_{d'} \sum_{i=1..m} w_i^{t+1} \phi_i(s,d',t+1). \end{aligned} \tag{3}$$

Equations (3) form a set of $|\mathcal{S}| \times |\mathcal{D}|$ linear equations for each time step $t \in T$, with variables $w_i^t, i = 1..m$. In general these systems are clearly over-constrained ($|\mathcal{S}| \times |\mathcal{D}| \gg m$), this is why we look for least-squares approximate solutions, instead of exact ones.

Dynamic programming. The dynamic programming part of LSDP comes from the fact that the systems are solved backwards for $t = H$ to 1, each solution vector $\{w_i^{t+1}\}_i$ being plugged into the system at time t.

Sampling of the complete least-squares system. At a given step t, system (3) is too large to build when $\mathcal{S}$ is factored, not to mention solving. Therefore, we suggest to *sample* this system, by considering only a subset of equations, corresponding to a subset of states $\mathcal{B} = \{(s,d,t)\} \subseteq \mathcal{S} \times \mathcal{D} \times T$, (called *batch* [29]). System (3) becomes:

$$\forall (s,d) \text{ s.t. } (s,d,t) \in \mathcal{B},$$

$$\sum_{i=1..m} w_i^t \phi_i(s,d,t) = r^t(s,d) + \sum_{s' \in Succ^t(s,t)} p^t(s'|s,d) V^w(s',t+1),$$
$$\text{where } V^w(s,t+1) = \max_{d'} \sum_{i=1..m} w_i^{t+1} \phi_i(s,d',t+1). \quad (4)$$

We propose to build the batch $\mathcal{B}$ from a finite set of simulated trajectories of the MDP, starting in s_1, obtained by simulating successive transitions. So doing, we have the guarantee that every 4-uplet $(s,d,t,s') \in \mathcal{B}$ effectively corresponds to a reachable configuration. At each transition of a trajectory (from (s,d,t) to s'), decision d is chosen according to the ε-greedy method: with probability $1-\varepsilon$ the decision is the one maximizing the current estimation Q^w and with probability ε the decision is chosen with uniform probability among all possible ones. Note that ε and the batch size are the only parameters to tune in LSDP.

6.2. LSDP algorithm in practice

From a set of weights we can straightforwardly derive the approximate Q-function and thus an approximation of the optimal policy and its value. Therefore, if δ^k is the current approximation of the optimal policy at iteration k of LSDP, iteration $k+1$ goes as follows:

- Construction of the new batch $\mathcal{B}^k$. It depends on δ^k since we apply the ε-greedy method to choose the decisions used to simulate transitions.
- Approximate resolution of (4) for each decision step, based on a least-squares estimation. The weights are updated and the corresponding policy δ^{temp} is evaluated.
- Updating of the policy: if the value of δ^{temp} is higher than that of δ^k, then $\delta^{k+1} = \delta^{temp}$. Otherwise $\delta^{k+1} = \delta^k$. The evaluation of δ^{temp} is obtained through MC simulations as the average over a large number of simulated trajectories of the total reward gained along these trajectories.

There is no guarantee that policy δ^{temp} improves the current policy (δ^k) in state s_1. This is the reason why we compare their values. If δ^{temp} does not improve δ^k, the iteration is once again initialized with δ^k. Since the batch generation is a stochastic procedure, the new batch will be different from $\mathcal{B}^k$, and we will obtain a new candidate policy δ^{temp}. This comparison step within one LSDP iteration guarantees that the successive policies returned by the algorithm are of increasing value. Simulation is used to estimate policy values, these estimations may well be incorrect but they hopefully preserve policies values ranking.

In practice, LSDP is initialized with a set of weights (one set per decision step in the MDP horizon). Then a maximum number of iterations is fixed, and when reached, the current policy is returned. See Figure 3 for a schematic representation of LSDP.

In the case where the resources constraints are not defined by a fixed number of sampling steps but by a maximal budget B, LSDP can still be applied. We simply define Q-functions and features as functions of b, the budget used so far, instead of functions of decision steps t performed so far. As a consequence, the sets of weights

are also indexed by the budget already spent. A trajectory is stopped when no action can be apply with the remaining budget. We will adopt this representation to solve the weed sampling problem (see Sections 7 and 8).

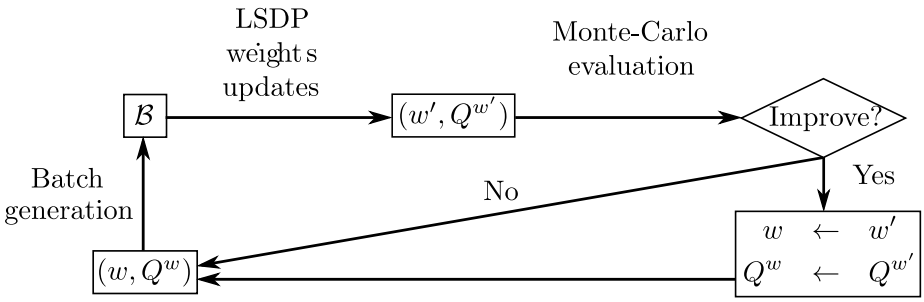

Figure 3: Schematic representation of the LSDP algorithm.

7. Application of LSDP to the OASMRF problem

In order to apply the LSDP algorithm to the OASMRF problem, we take into account the problem structure i) to define features ϕ_i and ii) to propose a time efficient batch construction method. It also requires iii) to be able to compute efficiently (in terms of time complexity) conditional marginals of the form $\mathbb{P}(x_i \mid x_A)$. These quantities are necessary to compute transition probabilities, to evaluate the final reward (the MPM value) and, as we will see, to compute the features. The solutions to these three points are described in Section 7.1. Together, they define one possible instantiation of LSDP for solving the OASMRF problem. We also present two alternative instantiations, based on different choices for the features definition or the quality of a trajectory (Section 7.2).

7.1. LSDP implementation for the OASMRF problem

Features choice. We chose to define one feature per variable in the MRF ($m = n$). The features definition is derived from the BP-max heuristic (see [26] and section 5). This heuristic consists in selecting for sampling, at each sampling step, the variables which remain the most uncertain. Uncertainty is measured by the maximal conditional marginal $\max_{x_i \in \Omega} \mathbb{P}(x_i \mid x_A)$: a low value indicates high uncertainty. This greedy heuristic can also be defined as the policy which maximizes at each decision step the following quantity $\sum_{i=1}^{n} \max_{x_i \in \Omega} \mathbb{P}(x_i \mid x_A)$, where $A \subseteq O$ is the set of sites sampled so far. Since when a site i has been sampled ($i \in A$), state x_i is known, we have $\max_{x_i \in \Omega} \mathbb{P}(x_i \mid x_A) = 1$ for $i \in A$. Therefore, the BP-max heuristic can be obtained as the greedy policy with respect to a parametrized Q-function Q^1 with the following features, and all weights equal to 1: $\forall i \in \{1, \ldots, n\}$,

$$\phi_i(s, d, t) \quad = \quad \big(1 - \mathbb{1}_{\{i=d\}}\big) \max_{x_i \in \Omega} \mathbb{P}\big(x_i \mid x_A\big) + \mathbb{1}_{\{i=d\}}. \tag{5}$$

We adopt definition (5) to define the LSDP features for the OASMRF problem. We initialize the LSDP algorithm with weights all equal to 1. Then, the LSDP algorithm performs successive updates in order to improve this initial set of weights.

Batch construction. Simulating trajectories in the OASMRF problem is costly since, for each transition, one has to simulate observations $x_{A^{t+1}}$ from the MRF conditional distribution $\mathbb{P}(x_{A^{t+1}} \mid x_{A^t})$. This requires to apply the Gibbs Sampling algorithm [10] a large number of times, which is rather costly, thus severely limiting the size and number of batches that can be constructed. However, larger batches can be constructed if we divide the construction into two phases. First, we simulate, off-line, a *batch of maps*, $\{x^1, \dots x^p\}$, from $\mathbb{P}$. It will be used for all iterations of the LSDP algorithm. The construction of this batch is done using Gibbs Sampling, and induces a single overhead cost (which can be large) for the whole algorithm. Then, at a given iteration k of LSDP, trajectories are easy to simulate: i) a map x is selected uniformly at random in the batch, ii) actions are chosen following the ε-greedy method with respect to the current policy, and iii) successive states s^t follow immediately by reading the values of the sampled variables corresponding to the current decision. The batch of states $\mathcal{B}$ is built as the set of all states encountered in all trajectories. This second phase of trajectories simulation is fast. Furthermore, simulated trajectories do not have to be stored (only the batch of maps does), thus saving much memory space. In addition, we can establish that this 2-step procedure is a valid method to simulate transitions of the MDP encoding of the OASMRF problem. More formally, we establish the following lemma

Lemma 1. *For a given action trajectory $(d^1, \dots, d^H)$, a state trajectory $(s^1, \dots, s^{H+1})$ simulated according to the following 2-step scheme has the same joint probability distribution as a trajectory simulated according to the OASMRF MDP model transition function:*

1. *Simulate a map x according to the joint distribution $\mathbb{P}(.)$.*
2. *Deduce iteratively the values $(s^1, \dots, s^{H+1})$ according to $s^1(i) = -1 \; \forall i \in O$ and:*

   $\forall t \in \{1, \dots, H\}$, $s^{t+1}(i) = s^t(i)$ *if* $d^t(i) = 0$ *and* $s^{t+1}(i) = d^t(i)x_i$ *else.*

   (We recall that a site is visited at most once during a trajectory).

A proof of this Lemma is given in the Appendix.

Approximation of $\mathbb{P}(x_i \mid x_A)$. The *Belief Propagation* (BP) algorithm [24] can be used to compute (approximately) $\mathbb{P}(x_i \mid x_A)$. However since this evaluation has to be performed a huge number of times, BP cannot be applied in practice. So we propose to use the distribution $\widetilde{\mathbb{P}}$ defined below as an approximation of $\mathbb{P}(x_i \mid x_A)$:

$$\widetilde{\mathbb{P}}(x_i \mid x_A) = \mathbb{P}^{BP}(x_i) + \sum_{j \in A} \Big[\mathbb{P}^{BP}(x_i \mid x_j) - \mathbb{P}^{BP}(x_i) \Big]. \tag{6}$$

This approximation does not necessarily belong to $[0, 1]$ but sums to one. It has the advantage to be fast to compute. Indeed, before running LSDP, all marginals and conditional marginals $\mathbb{P}^{BP}(x_i)$ and $\mathbb{P}^{BP}(x_i|x_j)$ are computed using BP, inducing a fixed

overhead computational cost. Then, within an iteration of LSDP, we can compute $\widetilde{\mathbb{P}}(x_i \mid x_A)$ in an incremental way since $\widetilde{\mathbb{P}}(x_i \mid x_A \cup x_j) = \widetilde{\mathbb{P}}(x_i \mid x_A) + \mathbb{P}^{BP}(x_i|x_j) - \mathbb{P}^{BP}(x_i)$. Our approximation is ad-hoc and we could have considered more sound methods to define an approximation of $\mathbb{P}(x_i \mid x_A)$ from the $\mathbb{P}^{BP}(x_i)$ and $\mathbb{P}^{BP}(x_i|x_j)$. Different options are discussed in [1]. In particular the authors pointed out the superiority of methods based on multiplication instead on addition. We did not explore this option since ours provided good empirical results and does not require any extra parameters estimation.

Example 3. *Weed sampling in a crop field.*
In our case study, budget is defined by a maximum time T_{max} than can be spent in the field for sampling. Each sampling step has a different duration since it depends on the location of the sampled site and on the observed density class. Therefore two trajectories of a given policy δ can have different lengths: they are stopped when any choice for an extra sampling step would lead to a total sampling time higher than T_{max}. When applying LSDP, we consider the case where b takes integer values and we solve (4) for every value b of time spent so far in the field encountered in the batch so far, instead of every decision step. For a given b, the subset of equations in (4) corresponds to the states in batch $\mathcal{B}^k$ reached after spending a time b in the field.

7.2. Two variants of LSDP for OASMRF

Static version of LSDP. It is possible, by changing the features definition, to design a static policy for the OASMRF problem. Here by static we mean that the choice of the next sample does not depend on the value of the variables observed in the previous sampling steps. It only depends on their locations. Therefore the set of sampled sites does not depend on the realization x of the hidden map and it can be computed in advance, before actually sampling the sites. Such a static policy can be obtained by considering the following definition for the features, $\forall i \in \{1, \ldots, n\}$,

$$\phi_i(s, d, b) = \mathbb{1}_{\{i=d\} \cup \{s_i \neq -1\}}.$$

The feature is equal to zero for all sites not sampled (at the current step or in previous ones) and 1 otherwise. In our experiments, we will compare this static policy to the above-defined version of LSDP.

Entropy based LSDP. The OASMRF problem and the LSDP algorithm have been described for a measure of sampling policy quality based on the MPM criterion (1). This choice is not arbitrary since with this definition the procedure used to restore the MRF state from a sample output and the procedure used to define the sampling policy quality rely on the same criterion. Still, other classical options can be considered to define the sampling policy quality. We could, for instance, define the OASMRF problem with an entropy-based criterion. In this case, since entropy has to be minimized, we define:

$$U\big(A, x_A\big) = -H(\mathbb{P}(X_R \mid x_A)) = \sum_{x_R} \mathbb{P}(x_R \mid x_A) \log(\mathbb{P}(x_R \mid x_A)),$$

with r^{H+1} defined accordingly. The steps of the LSDP algorithm would remain roughly unchanged with the entropy criterion except that the features definition should be adapted to: $\forall i \in \{1, \ldots, n\}$,

$$\phi_i(s, d, b) = -\big(1 - \mathbb{1}_{\{i=d\}}\big) H(\mathbb{P}(X_i \mid x_A)) + \mathbb{1}_{\{i=d\}},$$

where $A \subseteq O$ is the set of indices of the previously observed variables. Evaluating marginal entropy is not simpler than evaluating conditional marginals. In order to approximate these quantities we could again use approximation (6).

Note that the entropy criterion does not provide a rule to estimate the variables X_R from a sample output. This reconstruction step still has to be performed using MPM or MAP methods.

8. EXPERIMENTAL EVALUATION

We present simulated sampling problems and one real problem on weed sampling in a crop field to illustrate the gain of using LSDP instead of classical heuristics or RL-based solution algorithms. We compared LSDP to the random heuristic, the LSDP-static policy, the BP-max policy, TD(λ) with tabular representation of the Q-function, and LSPI. LSPI and LSDP were implemented with the same features definition and were run with $\epsilon = 0.9$. We also compared LSDP to a greedy algorithm based on the *Mutual Information* (MI) criterion [16].

The OASMRF problem considered is the following. The graph G is a regular grid and $R = O = V$. One variable is observed at each decision step ($L = 1$) and sampling costs are null on the three first sets of experiments. We considered the following Potts model distribution: $\forall\, x \in \{0, 1\}^n$

$$\mathbb{P}(x) \propto \exp\left(\frac{1}{2} \sum_{(i,j)\in E} \mathbb{1}_{\{x_i = x_j\}}\right).$$

4×4 grid. This small problem was introduced in the experiments since we were able to compute the corresponding optimal policy, using the backward induction algorithm (see Section 5), and the exact value of any policy. TD(λ) was run with $\lambda = 0.1$, using the ε-greedy method for action choice ($\epsilon = 0.1$). It was run using 675000 simulated state-action trajectories, in order to reach convergence. To be comparable, we ran LSDP and LSPI with a batch of 100 maps and 6750 iterations (in practice a few hundred iterations are enough). For LSDP the value of the policy obtained at the last iteration of the algorithm was returned, and for LSPI the value of the best policy among all iterations was returned.

The first conclusion is that the absolute difference between the values of all policies is small: an absolute increase of the percentages of 2.2 at most. We also compared the policies in terms of normalised gain compared to the random one δ_R (Figure 4): the score of a given policy δ is defined as $score1(\delta) = \frac{V(\delta) - V(\delta_R)}{V(\delta^*) - V(\delta_R)}$.

Among RL algorithms, TD(λ) is the best and LSDP gives very similar results. In comparison, LSPI shows a poor behaviour, always returning dominated policies. Surprisingly the relative values of the MI and LSPI policies decrease with the number of

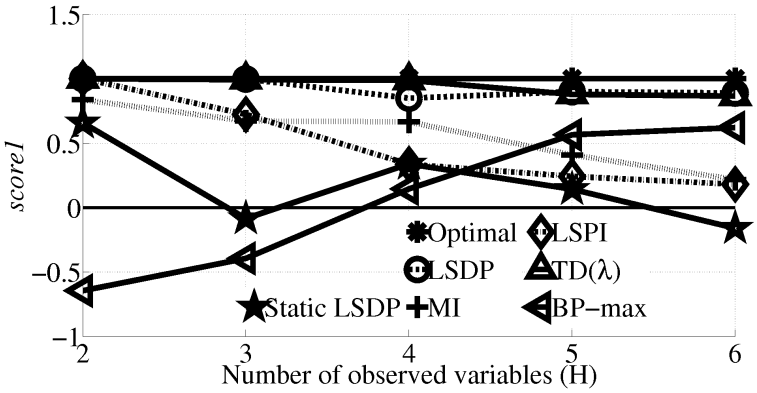

Figure 4: OASMRF problem with 16 variables: $score1$ of LSDP and classical one-step-look-ahead and RL-based heuristic policies. A policy with $score1$ equal to 0 is a policy with the same value as the random policy.

observed variables, while the opposite behavior is observed for the BP-max heuristic. The poor performance of the BP-max heuristic with small sample size is explained by the fact that with few observed sites, all sites have similar marginal probabilities. In that situation we arbitrarily choose the site to sample as the one with the lowest index in V.

10×10 grid. For this problem size, only LSDP, LSDP-static, LSPI, BP-max and random policy can be computed. For LSDP, LSDP-static and LSPI we used a batch size of 1000 maps and 1000 iterations. The value of a policy was estimated by Monte Carlo approximation. We modified $score1(\delta)$ into $score2(\delta) = \frac{V(\delta)-V(\delta_R)}{|V(\delta_{BP-max})-V(\delta_R)|}$: since the value of an optimal policy cannot be computed, δ_{BP-max} serves as a reference. Results are displayed on Figure 5.

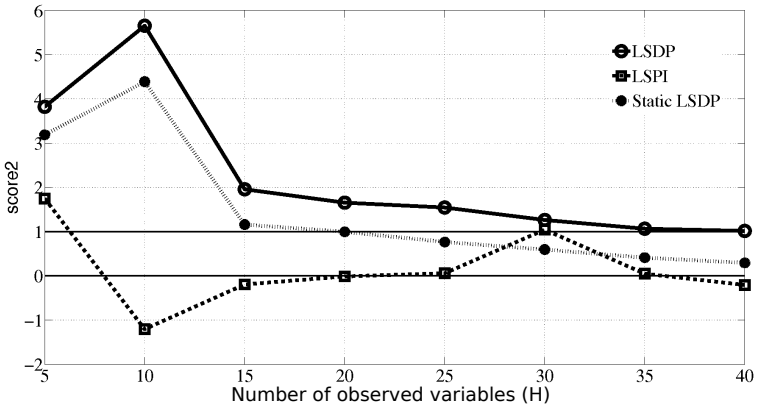

Figure 5: OASMRF problem with 100 variables: $score2$ of LSDP, LSDP-static and LSPI policies. A policy with $score2$ equal to 0 (resp. 1) is a policy with the same value as the random (resp. BP-max) policy.

We observed again the poor performance of the LSPI algorithm (dominated by the random policy for $H = 10$ to 20). On the contrary, LSDP performs quite better than the BP-max heuristic for small sample sizes. LSDP also performs better than LSPI, in terms of computation time: for $H = 40$, an iteration takes about 7 seconds for LSDP, 77 seconds for LSPI. For these reasons (poor performance, high computation time), we did not consider LSPI in the following experiments.

The LSDP-static policy also leads to an improvement compared to BP-max, but lower than with LSDP: this example and the previous one demonstrate the interest of looking for adaptive policies.

Constrained moves problem. We compared LSDP, BP-max and random policies on a more realistic sampling problem, involving constrained moves on the grid for observing sites. The agent starts by sampling the site at the top-left corner of the grid. Then, after having observed a site, the agent can only move to distance-2 sites for the next observation..

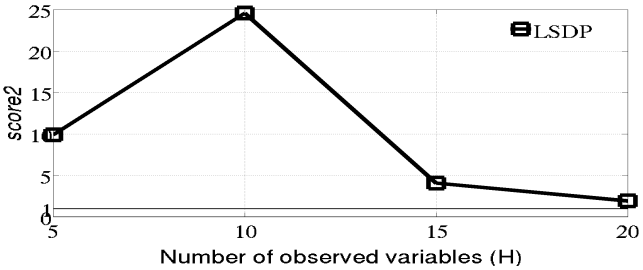

Figure 6: Constrained moves problem with 100 variables: $score2$ of LSDP policy. A policy with $score2$ equal to 0 (resp. 1) is a policy with same value as the random (resp. BP-max) policy.

We again observed that the absolute difference between all policies remained small (for $H = 10$, the value of the LSDP policy is 61.7 while the value of the BP-max policy is 59.4). LSPI showed the same poor behaviour than in the previous experiment. As we expected, the gain provided by LSDP in terms of relative improvement of the random policy ($H \leq 20$, see Figure 6) is significant when the sample size is small (Figure 6).

Sampling under cost constraints With this set of experiments we introduced distincts costs values $S_C(i, x_i)$ and we considered the problem of maximising the restoration quality under the constraint of a fixed allocated budget B. We considered three different cost functions $S_C(A, x_A)$. For the type I and type II, cost depends only on the site location. With cost I, the sampling cost increases with the distance to the grid boundary, while with cost II, we have two different costs in the two diagonals (see Figure 7). With the type III cost, we consider a function S_C which depends only on the value of the observation: $S_C(i, x_i) = 2$ if $x_i = 1$ and 1 otherwise. We ran the LSDP, BP-max and random policy on a 20 $\times$ 10 grid and for a budget $B = 38$. For LSDP we used a batch of size 4000 or 2000, and 1000 iterations. Results in terms of policy values and numbers of sites sampled are presented in Table 2. For the three types of cost function, one can observe that the ranking of the three policies values is always

	Type I cost		Type II cost		Type III cost	
Policy	Value	sampled sites	Value	sampled sites	Value	sampled sites
LSDP	64.80	27.3 (2.5)	63.6	22.8 (0.7)	65.4	25.6 (1.7)
BP-max	61.77	19 (1.9)	60.4	15.8 (1.9)	64.7	25.6 (1.8)
Random	60.27	26.65 (2.8)	59.7	15.6 (2.3)	63.7	25.6 (1.9)

Table 2: Values and mean number of visited sites under different configurations of cost constraints, for the LSDP, BP-max and random policies. Values between parentheses are standard deviations for the mean number of visited sites.

LSDP $>$ BP-max $>$ random. The LSDP policy distributes the budget B in a way that enables to sample more sites than with BP-max.

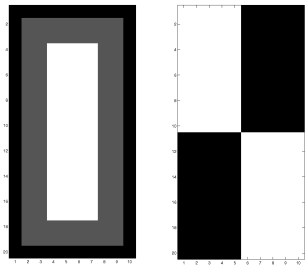

Figure 7: Left: type I repartition of costs, costs are respectively of 1, 2 and 4 for black, grey and white sites. Right: type II repartition of cost, costs are respectively of 1 and 4 for black and white sites.

Weed sampling in a crop field under time constraint. We also applied LSDP to the problem of designing adaptive policies for weed sampling in a crop field, described in Section 2. A spring barley field has been divided into a regular grid of 13×13 quadrats of $12.96m^2$ area each. The density classes of the weed species *Galium Aparine* was recorded on each quadrat to construct the vector x. The observation x_i in quadrat i is the weed density class and belongs to one of the three following classes: 0 (no weeds), 1 (less than one plant per square meter), 2 (between 1 and 3 plants per square meter). We considered different MRF models corresponding to different properties and we selected the model with the highest BIC value [13]. This model was an anisotropic Potts model with external field:

$$\mathbb{P}_\beta\big(x\big) \propto \exp\left(\sum_{i\in V} \alpha_{x_i} + \beta_t \sum_{(i,j)\in E_t} \mathbb{1}_{\{x_i=x_j\}} + \beta_o \sum_{(i,j)\in E_o} \mathbb{1}_{\{x_i=x_j\}}\right).$$

Subsets E_t and E_s respectively represent the subsets of edges in tillage direction and in the orthogonal direction, since tillage can be responsible for a difference of spatial correlation between these two directions. The estimated parameters were: $(\alpha_0, \alpha_1, \alpha_2) = (0, -0.03, -3.58)$ and $(\beta_t, \beta_o) = (0.71, 0.12)$. They have been estimated from the observed vector x by maximisation of the pseudo-likelihood approximation [4]. The

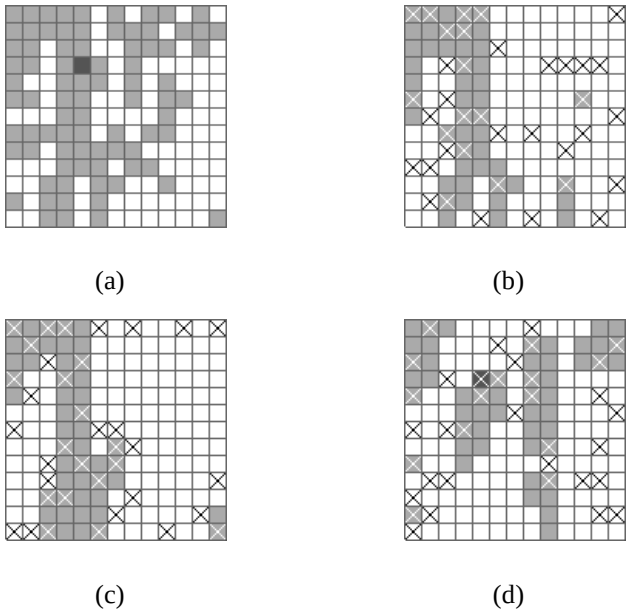

Figure 8: Sampling policies for weeds mapping. (a) true density map of *Galium Aparine*, and MPM reconstruction based on (b) LSDP sampling policy, (c) BP-max sampling policy, (d) a random policy. Sampled quadrats are marked by a x. White, gray and black quadrats correspond respectively to density classes 0, 1 and 2.

cost function $S_C(i, x_i)$ represents the time needed, in seconds, for density class assessment in a quadrat (a site of the MRF). To define this function, we used a regression model based on factors identified as the most relevant by experts. Here the cost value $S_C(i, x_i)$ both depends on the quadrat location i and the observed density class x_i. The influence of the quadrat location is due to the influence of the number and density classes of the other weeds species present in quadrat i. These data were also recorded when sampling *Galium Aparine*. The time needed for abundance estimation increases slowly with the density class. The minimum and maximum observation durations are respectively 190 seconds and 360 seconds. The mean observation duration is about 300 seconds with 35 seconds standard deviation. For this experiment, sampling budget was fixed to 9000 seconds (2h30).

We applied the three policies LSDP, BP-max and random to sample and reconstruct the original weed abundance map used to build the MRF model. LSDP was applied with a batch size of 4000 and 1000 iterations. The true density map of *Galium Aparine* and maps estimated from sample outputs provided by the three policies are presented on Figure 8. Note that in the true map there is only one quadrat in class 2. This is not rare in weed maps, but whatever the sampling method, it is very unlikely that a MPM (or MAP) reconstruction classifies correctly this particular quadrat.

Once again the two adaptive policies increase the quality of the reconstructed map compared to the random one: the numbers of quadrats where the density classes are

well estimated are 125, 124 and 111 respectively, for policies LSDP, BP-max and random. The difference between the two restorations provided by the adaptive policies is small. However, the corresponding explorations of the crop field are different. As an illustration, Figures 9 and 10 display, for the two adaptive polices, the results of the successive sampling steps, grouped by 6 successively sampled quadrats. One can see from these figures that the sampled quadrats are more scattered across space with the LSDP policy. And this is the case even for the first sampling steps. On the contrary, the BP-max policy concentrates most of the sampling in the area where the weed is observed at density class 1. Then, since the LSDP policy takes into account the remaining budget to decide which quadrat to sample next, it was able to observe one more quadrat than the BP-max policy (38 instead of 37).

9. CONCLUSION

In this article, we have provided a factored MDP model to represent problems of optimal adaptive sampling of spatial processes expressed in the MRF framework. Our second contribution is a generic *batch mode RL* algorithm, LSDP, which can be applied to any large state-space finite-horizon MDP problem, as soon as the MDP model is known explicitly. Then, our last contribution is an experimental evaluation of the LSDP approach for solving the OASMRF problem. Our experimental work enables us to draw the following conclusions. First, in small problems where the optimal policy can be computed, we notice that the performance of a purely random policy is quite close to that of the optimal one. This seems to also hold for larger problems, where the estimated value of the random policy remains close to that of the LSDP policy. However, in real-life applications of sampling for mapping, small errors in the reconstruction of maps can lead to a significant increase in management costs (think of imperfect mapping and eradication of invasive species, leading to future catastrophic outbreaks). Second, for large problems, non-parameterized RL approaches (such as TD(λ)) are too computationally intensive to apply, and the LSPI approach does not perform well. On the contrary, both BP-max heuristic and the LSDP algorithm provide good results (provided that the sampling budget is large enough, as far as BP-max is concerned). BP-max is less computationally expensive to apply than LSDP. However, its main drawback is that the choice of the sample does not take into account its cost. The budget constraint can only be used to decide when to stop a sampling trajectory. In contrast, LSDP can handle cost functions and our experiments show that when sampling costs are nonhomogeneous the superiority of LSDP over BP-max and random policies is increased.

Our work has similarities with other recent approaches [15, 19, 25, 26] to sampling in graphical models. Most of these approaches combine heuristic estimations of policy values with greedy or dynamic programming approaches. [19], in particular, have defined a dynamic programming approach to decision in graphical models, similar to the one presented in [26] and in the present work. Their objective is, as ours, to define a sampling strategy, but their reward function is simpler to compute than a reconstruction quality in MRF (they consider an additive aggregation of simple reward functions). However, they face similar problem to ours, with a decision-tree too large to explore completely to build an adaptive sampling policy. While LSDP tackles this

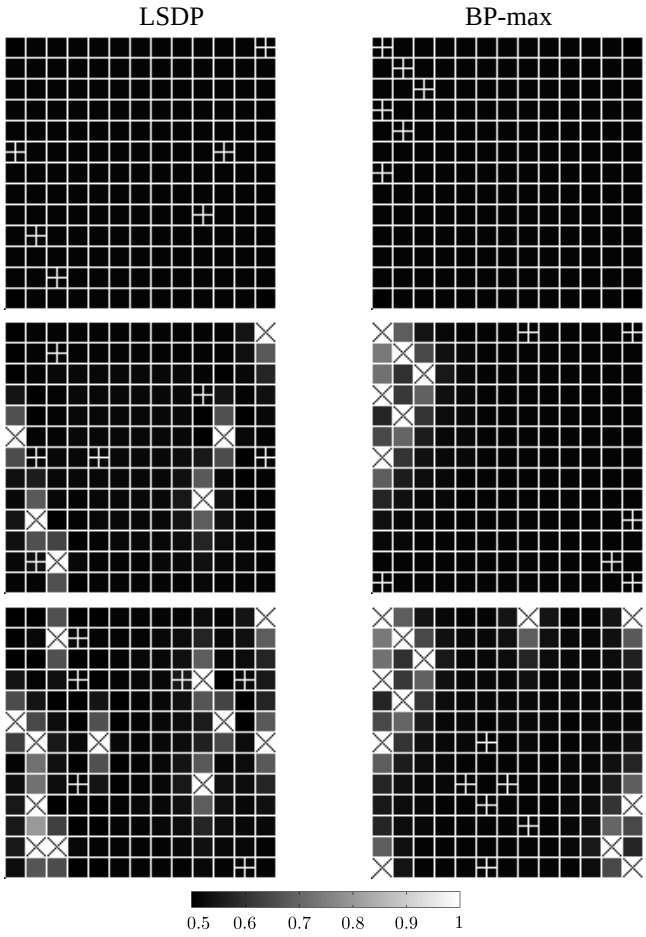

Figure 9: Locations of sampled quadrats and max-marginal values for LSDP (left column) and BP-max (right column) policies. From top to bottom, each figure respectively shows the cumulated samples 1 to 6, 7 to 12 and 13 to 18. Previously sampled quadrats are marked with a $\times$ and the 6 new ones with a $+$. The grey scale indicates the remaining uncertainty before the 6 new sampling steps: black (resp. white) encodes a max-marginal equals to 1/2 (resp. 1).

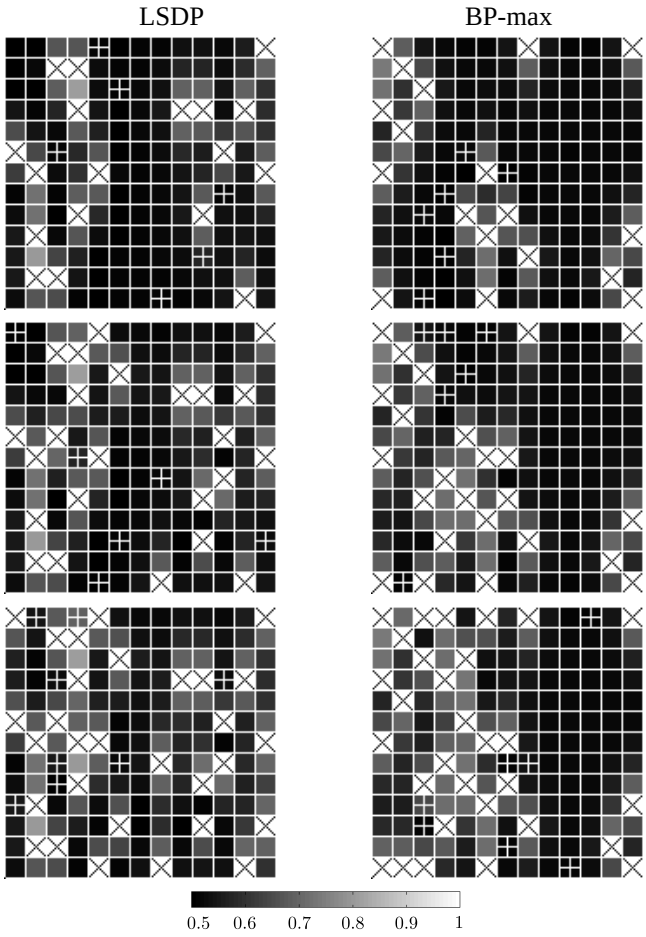

Figure 10: Locations of sampled quadrats and max-marginal values for LSDP (left column) and BP-max (right column) policies. From top to bottom, each figure respectively shows the cumulated samples 19 to 24, 25 to 30 and 31 to 37 (for BP-max) or 38 (LSDP). Previously sampled quadrats are marked with a $\times$ and the 6 new ones with a $+$. The grey scale indicates the remaining uncertainty before the 6 new sampling steps: black (resp. white) encodes a max-marginal equals to 1/2 (resp. 1).

problem by "sampling" complete trajectories of the full decision tree, [19] suggest to explore a bounded-depth subtree (of reasonable size) of the full decision tree, with heuristic values attached to the leaves. Theoretical and experimental comparisons of both approaches are left for further research.

This work opens several directions for future work: on the problem of sampling in spatial random fields in one hand, and on more general problems of sequential decision under uncertainty. Regarding the framework and algorithm we proposed for spatial sampling, a first possible extension would be to consider other definitions of sample quality measures. In this paper, the measure used to illustrate the approach is the MPM value. However, the MDP encoding and the application of LSDP do not crucially depend on the quality measure definition. Other criteria, such as MAP, or Entropy should be explored. It would probably require to define new features, as we have illustrated for the entropy case, and belief propagation algorithms could still be used to compute approximately MAP or entropy values.

We largely discussed the different options to introduce cost constraints in the optimal sampling problem. We have modelled our sampling problem as a problem of optimising reconstruction quality, under sampling budget constraint. However, one could, dually, be interested in finding sampling policies achieving a minimum reconstruction quality threshold, while minimising the sampling cost. An MDP encoding of this problem is still possible and the LSDP algorithm could be applied. It would require an MPM evaluation at every sampling step to check if the minimal quality is reached, but this can be evaluated approximately based on our time efficient approximation of the conditional marginals. Other forms of sampling cost could also be discussed: these could be more general than the ones we have considered in the paper. These could be linked, for example, to a maximum sampling trajectory duration, modelled as a sum of transitions (s, a, s') costs. Finally, even the choice of a MRF to model map uncertainty can be challenged, while keeping the approach we proposed. One could easily adapt the principles of our approach to continuous space models, provided that the number of potential sampling locations be finite. In a MRF, each variable typically take values in a finite set of small size. We could consider applying LSDP to problems with larger (but still finite) domains, when counts data should be modelled. The only requirement would be to be able to efficiently compute conditional marginals and simulate full maps. If the domain of the variable to map is continuous, this rises the more complex question of the definition of MDP on continuous state space.

Then, as we already mentioned, the LSDP algorithm is not specific to the resolution of the optimal sampling problem. One important contribution of this work is a new model-based RL algorithm for large size finite-horizon MDP. This means that it can be applied to solve problems of sequential decision under uncertainly where the state and/or decision space are/is large and factored (eg. invasive species control, biodiversity conservation, weeds management, ...). MDP formalisms already exist to model the control of spatial processes in time: Factored MDP (FMDP) [12] and Graph-based MDP (GMDP) [30], for example. The structure of these MDPs shares numerous common points with the MDP model of the OASMRF problem. Clearly, the LSDP approach could be adapted to approximately solve FMDP or GMDP problems, when the horizon is finite.

Acknowledgement

We would like to thank Alain Dutech, Bruno Scherrer and Bruno Zanuttini for fruitful discussions on latest Reinforcement Learning advances, as well as Sabrina Gaba whose research on weeds management motivated this work. We are grateful to Fabrice Dessaint and Amélie Slaski who kindly provided the data set used for the illustration of LSDP on a weed sampling problem. This work was funded by the ANR project LARDONS under grant ANR-10-BLAN-0215.

Appendix

Proof of Proposition 1:

First, let us define $h^1 = ((\emptyset, \emptyset))$ and $\forall t = 2, ..., H$, $h^t = ((A^1, x_{A^1}), \ldots, (A^{t-1}, x_{A^{t-1}}))$. From any history h^t, a unique MDP state $s^t(h^t)$ can be defined as $s^1(h^1) = (\emptyset, \emptyset)$ and $\forall t = 2, ..., H$, $s^t(h^t) = \left(\cup_{k=1}^{t-1} A^k, x_{\cup_{k=1}^{t-1} A^k} \right)$. Then, we define the following transformation ϕ from the set of MDP policies to the set of OASMRF policies. For π, a MDP policy, $\delta = \phi(\pi)$ is defined as: for any $t = 1 \ldots H$ and any reachable trajectory h^t, $\delta^t(h^t) = \pi^t(s^t(h^t))$.

(i) We first show that $V^\pi((\emptyset, \emptyset), 1) = V(\phi(\pi))$. Indeed, we recall that

$$\begin{aligned} V^\pi((\emptyset, \emptyset), 1) &= \mathbb{E}_\pi\big[\sum_{t=1}^{H+1} r^t \mid s^1 = (\emptyset, \emptyset)\big] \\ &= \sum_{s^2, \ldots, s^{H+1}} \mathbf{P}(s^2, \ldots, s^{H+1} \mid \pi, s^1 = (\emptyset, \emptyset)) \Big[\sum_{t=1}^{H} r^t(\pi^t(s^t)) + r^{H+1}(s^{H+1}) \Big], \end{aligned}$$

where $\mathbf{P}(s^2, \ldots, s^{H+1} \mid \pi, s^1)$ is the probability of the state trajectory $(s^2, \ldots, s^{H+1})$, starting from s^1 and following policy π. Note that for any "feasible" MDP state trajectory $s^1, \ldots, s^{H+1}$ we can define a unique history $h^{H+1} = ((A^1, x_{A^1}), \ldots, (A^H, x_{A^H}))$, where A^t is the set of vertices involved in s^{t+1} and not in s^t. Then:

$$\mathbf{P}(s^2, \ldots, s^{H+1} \mid \pi, s^1) = \begin{cases} 0 \text{ if state trajectory not reachable,} \\ \mathbb{P}(x_A \mid) \text{ otherwise.} \end{cases}$$

with $A = \cup_{t=1}^{H} A^t$. In addition, we have that: $r^t(\pi(s^t)) = 0$ and $r^{H+1}(s^{H+1}) = \sum_{r \in R} \max_{x_r \in \Omega} \{\mathbb{P}(x_r \mid x_A)\}$. Finally

$$\begin{aligned} V^\pi((\emptyset, \emptyset), 1) &= \sum_{h^{H+1} \in \tau_{\phi(\pi)}} \mathbb{P}(x_A \mid) \Big[-\alpha \sum_{t=1}^{H} \sum_{a \in A^t} c_a + \sum_{r \in R} \max_{x_r \in \Omega} \{\mathbb{P}(x_r \mid x_A)\} \Big] \\ &= \sum_{h^{H+1} \in \tau_{\phi(\pi)}} \mathbb{P}(x_A) U(A, x_A) \\ &= V(\phi(\pi)). \end{aligned}$$

(ii) Then, we prove by backwards induction that an optimal policy δ^* for the OASMRF problem can be defined, which prescribes successive samples independently of the order of past observations. Let us consider $\delta^{*,H}$ first.

$$\delta^{*,H}((A^1, x_{A^1}), \ldots, (A^{H-1}, x_{A^{H-1}})) = \arg\max_{A^H} \sum_{x_{A^H}} \mathbb{P}(x_{A^H} | x_{A^1}, \ldots, x_{A^{H-1}}) U(A, x_A),$$

where $A = A^1 \cup \ldots \cup A^H$. Both $\mathbb{P}(x_{A^H} | x_{A^1}, \ldots, x_{A^{H-1}})$ and $U(A, x_A)$ do not depend on the order of observations $x_{A^1}, \ldots, x_{A^{H-1}}$. Thus, $\delta^{*,H}$ does not depend on the order of its arguments.
Now, at time $h = H - 1$:

$$\delta^{*,H-1}((A^1, x_{A^1}), \ldots, (A^{H-2}, x_{A^{H-2}})) = \arg\max_{A^{H-1}} \sum_{x_{A^{H-1}}}$$

$$\sum_{x_{A^H}} \mathbb{P}(x_{A^{H-1}}, x_{A^H} | x_{A^1}, \ldots, x_{A^{H-2}}, \delta^{*,H}(\ldots)) U(A, x_A).$$

Since $\delta^{*,H}$ does not depend on the order of its arguments, $\delta^{*,H-1}$ is also independent of the order of its arguments $x_{A^1}, \ldots, x_{A^{H-2}}$.
Following the same reasoning for $h = H-2, \ldots, 1$, we prove that an optimal policy δ^* can be computed, which prescribes samples independently of past observations. This result implies that we can limit the search for optimal policies of the OASMRF problem to policies δ prescribing actions which do not depend on the order of observations.

(iii) Let us now consider a given policy δ of the OASMRF in our limited search space. We can derive a policy π of the corresponding MDP model. The construction is also by induction: $\pi^1((\emptyset, \emptyset)) = \delta^1$, and for $t = 2$ to H and a reachable state s^t we define a history $((A^1, x_{A^1}), \ldots, (A^{t-1}, x_{A^{t-1}}))$ of size t-1, where the order in which observations are made are choosen arbitrarily, and we set $\pi(s^t) = \delta^t((A^1, x_{A^1}), \ldots, (A^{t-1}, x_{A^{t-1}}))$. With this procedure, π is defined only for states s^t reachable from policy δ. For other states, the policy is set to an arbitrary decision (the value of π will not depend on this choice since the corresponding state will never be reached). Let us call μ this transformation from a OASMRF policy to a MDP policy. Following the same reasoning as in (i), we can easily show that $V^{\mu(\delta)}((\emptyset, \emptyset), 1) = V(\delta)$.

(iv) Finally, let π^* be the optimal policy of the MDP model of the OASMRF problem:

$$V^{\pi^*}(s, t) \geq V^{\pi}(s, t) \qquad \forall \pi, s, t$$

Therefore the policy $\phi(\pi^*)$ is optimal for the OASMRF problem. Indeed, let δ be a given policy of the OASMRF problem (with the property of independence on the observations order) and $\mu(\delta)$ the corresponding policy of the MDP model. We have that

$$V^{\pi^*}((\emptyset, \emptyset), 1) \geq V^{\mu(\delta)}((\emptyset, \emptyset), 1),$$

and since $V^{\pi^*}((\emptyset, \emptyset), 1) = V(\phi(\pi^*))$ and $V^{\mu(\delta)}((\emptyset, \emptyset), 1) = V(\delta)$, we obtain $V(\phi(\pi^*)) \geq V(\delta)$. This establishes Proposition 1.

Proof of Lemma 1:
For a given action trajectory $(d^1, \ldots, d^H)$, let us consider a state trajectory $(s^1, \ldots, s^{H+1})$ simulated according to the following 2-step scheme.

1. Simulate a map x according to the joint distribution $\mathbb{P}(.)$.
2. Deduce iteratively the values $(s^1, \ldots, s^{H+1})$ according to $s^1(i) = 0,\ \forall i \in O$ and:

$$\forall t \in \{1, \ldots, H\},\ s^{t+1}(i) = s^t(i) + d^t(i)x_i. \tag{7}$$

We have that

$$\mathbf{P}(s^1, \ldots, s^{H+1} \mid d^1, \ldots, d^H) \quad = \quad \sum_{x_V \in \Omega^n} \mathbb{P}(x_V)\mathbf{P}(s^1, \ldots, s^{H+1} \mid x_V, d^1, \ldots, d^H).$$

The probability $\mathbf{P}(s^1, \ldots, s^{H+1} \mid x_V, d^1, \ldots, d^H)$ is either equal to zero or to 1, since only one state trajectory can be reached from x_V and $(d^1, \ldots, d^H)$ according to (7). Furthermore, given $(d^1, \ldots, d^H)$, the state trajectory $(s^1, \ldots, s^{H+1})$ can be reached from any configuration x_V which agrees with the observations of this state trajectory on the subset A of sites visited by the action trajectory $(d^1, \ldots, d^H)$. Thus, if x'_A is the set of observations collected on A along the state trajectory $(s^1, \ldots, s^{H+1})$

$$\mathbf{P}(s^1, \ldots, s^{H+1} \mid d^1, \ldots, d^H) = \sum_{x_V \in \Omega^n} \mathbb{P}(x_V)\mathbb{1}_{\{x_A = x'_A\}},$$

which by definition is equal to $\mathbb{P}(x'_A)$.

Let us now evaluate the propability to observe the same state trajectory $(s^1, \ldots, s^{H+1})$, given $(d^1, \ldots, d^H)$, when simulating according to the OASMRF MDP transition function:

$$\mathbf{P}(s^1, \ldots, s^{H+1} \mid d^1, \ldots, d^H) = \mathbb{P}(x'_{d^1}) \prod_{t=2}^{H} \mathbb{P}(x'_{d^t} \mid x'_{d^{t-1}}, \ldots, x'_{d^1}).$$

Using the Bayes rule, one can see that $\mathbb{P}(x'_{d^1}) \prod_{t=2}^{H} \mathbb{P}(x'_{d^t} \mid x'_{d^{t-1}}, \ldots x'_{d^1})$ is exactly $\mathbb{P}(x'_{d^1}, \ldots, x'_{d^H})$, which is equal to $\mathbb{P}(x'_A)$.

Therefore, with the two simulation schemes, for a given action trajectory $(d^1, \ldots, d^H)$ the same state trajectories can be reached (those where visited sites are coherent with the actions) and each state trajectory has the same probability in both schemes.

References

[1] D. Allard, A. Communian, and P. Renard. Probability aggregation methods in geoscience. *Mathematical Geosciences*, 44(5):545–581, 2012.

[2] A. Antos, C. Szepesvarí, and R. Munos. Learning near-optimal policies with Bellman residual minimization based fitted policy iteration and a single sample path. *Machine Learning*, 71(1):89–129, 2008.

[3] D.P. Bertsekas. *Dynamic Programming and Optimal Control, vol II*. Athena Scientific, 4th edition, 2012.

[4] J. Besag. Statistical analysis of non-lattice data. *The Statistician*, 24(3):179–195, 1975.

[5] M. Bonneau, N. Peyrard, and R. Sabbadin. A reinforcement learning algorithm for sampling design in Markov random fields. In *European Conference on Artificial Intelligence*, 2012.

[6] R.D. Cousens, R.W. Brown, A.B. McBratney, B. Whelan, and M. Moerkerk. Sampling strategy is important for producing weed maps : a case study using kriging. *Weed science*, 50(4):542–546, 2002.

[7] J. de Gruijter, D. Brus, M. Bierkens, and K. Knotters. *Sampling for Natural Resource Monitoring*. Springer, 2006.

[8] D. Ernst, P. Geurts, and L. Wehenkel. Tree-based batch mode reinforcement learning. *Journal of Machine Learning Research*, 6:503–556, 2005.

[9] E. Evangelou and Z. Zhu. Optimal predictive design augmentation for spatial generalised linear mixed models. *Journal of Statistical Planning and Inference*, 142(12):3242–3253, 2012.

[10] S. Geman and D. Geman. Stochastic relaxation, Gibbs distribution, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:721–741, 1984.

[11] R.H. Gibson, I.L. Nelson, G.W. Hopkins, B.J. Hamlett, and J. Memmott. Pollinator webs, plant communities and the conservation of rare plants: arable weeds as a case study. *Journal of applied ecology*, 43(2):246–257, 2006.

[12] C. Guestrin, D. Koller, R. Parr, and S. Venkataraman. Efficient solution algorithms for factored MDPs. *Journal of Artificial Intelligence Research*, 19:399–468, 2003.

[13] R. Kass and A. Raftery. Bayes factors. *Journal of the American Statisticial Association*, 90:733–795, 1995.

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

[15] A. Krause and C. Guestrin. Optimal value of information in graphical models. *Journal of Artificial Intelligence Research*, 35:557–591, 2009.

[16] A. Krause, A. Singh, and C. Guestrin. Near-optimal sensor placements in Gaussian processes: theory, efficient algorithms and empirical studies. *Journal of Machine Learning Research*, 9:235–284, 2008.

[17] M. Lagoudakis and R. Parr. Least-squares policy iteration. *Journal of Machine Learning Research*, 4:1107–1149, 2003.

[18] H.R. Maei and R.S. Sutton. A general gradient algorithm for temporal-difference prediction learning with eligibility traces. In *International Conference on Artificial General Intelligence*, 2010.

[19] G Martinelli, J Eidsvik, and R Hauge. Dynamic decision making for graphical models applied to oil exploration. Technical report, Department of Mathematical Sciences, Norwegian University of Science and Technology, 2011.

[20] WG Müller. *Collecting spatial Data*. Springer Verlag, 2007. 3rd ed.

[21] R. Munos and C. and Szepesvarí. Finite-time bounds for fitted value iteration. *Journal of Machine Learning Research*, 9:815–857, 2008.

[22] E.C. Oerke. Crop losses to pests. *Journal of Agricultural Science*, 144:31–43, 2006.

[23] D. Ormoneit and S. Sen. Kernel-based reinforcement learning. *Machine Learning*, 49:161–178, 2002.

[24] J. Pearl. *Probabilistic Reasonning in Intelligent Systems*. Morgan Kaufmann, 1988.

[25] N. Peyrard, R. Sabbadin, and U. F. Niaz. Decision-theoretic optimal sampling with hidden Markov random fields. In *European Conference of Artificial Intelligence*, 2010.

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

[27] W. B. Powell. *Approximate Dynamic Programming: Solving the Curses of Dimensionality (Wiley Series in Probability and Statistics)*. Wiley-Interscience, 2007.

[28] M. Puterman. *Markov Decision Processes : Discrete Stochastic Dynamic Programming*. John Wiley & Sons, 1994.

[29] E. Rachelson, F. Schnitzler, and L. Wehenkel ans D. Ernst. Optimal sample selection for batch-mode reinforcement learning. In *International Conference on Agent and Artificial Inteligence*, 2011.

[30] R. Sabbadin, N. Peyrard, and N. Forsell. A framework and a mean field algorithm for the local control of spatial processes. *International Journal of Approximate Reasoning*, 53(1):66–86, 2012.

[31] R. S. Sutton and A.G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 1998.

[32] R.S. Sutton, C. Szepesvari, and H.R. Maei. A convergent O(n) temporal difference algorithm for off-policy learning with linear function approximation. In *Neural Information Processing Systems Conference*, 2008.

[33] S. Thompson and G. Seber. *Adaptive sampling*. Series in Probability and Statistics. Wiley, 1996.

[34] L.J. Wiles. Sampling to make maps for site specific weed management. *Weed science*, 53(2):228–235, 2005.