GMDPtoolbox: a Matlab library for solving Graph-based Markov Decision Processes

Cros Marie-Josée[1], **Peyrard Nathalie**[2], **Sabbadin Régis**[3]

1. *INRA, UR875, MIAT, Université de Toulouse, France*
   Marie-Josee.Cros@toulouse.inra.fr
2. *INRA, UR875, MIAT, Université de Toulouse, France*
   Nathalie.Peyrard@toulouse.inra.fr
3. *INRA, UR875, MIAT, Université de Toulouse, France*
   Regis.Sabbadin@toulouse.inra.fr

ABSTRACT. *Systems management in ecology or agriculture is complex because several entities in interaction must be managed together with a long term objective. Finding an optimal (or at least a good) policy to govern these large systems is still a challenge in practice. Graph-based Markov Decision Processes (GMDPs) form a suitable tool for modelling and solving such structured problems of sequential decision under uncertainty. In this article we introduce GMDPtoolbox: a Matlab library dedicated to the GMDP framework. The toolbox allows to easily represent a problem as a GMDP, to solve it (e.g. finding a "good" local policy) and finally to analyze a policy or compare its performance with human-built policies, like expert ones.*

RÉSUMÉ. *La gestion de systèmes en écologie et agronomie est une tâche complexe du fait de la nécessaire prise en compte simultanée de différentes entités en intéraction et d'objectifs définis à long terme. En pratique, trouver une politique optimale (ou au moins satisfaisante) est encore un challenge pour ces problèmes de grande taille. Les Processus Décisionnels de Markov sur Graphe (PDMG) fournissent un cadre adapté pour modéliser et résoudre de tels problèmes structurés de décision séquentielle sous incertitude. Dans cet aticle, nous présentons la GMDPtoolbox : une boite à outils Matlab dédiée au cadre des PDMG. La boite à outils permet de représenter facilement un problème, de trouver une solution sous forme d'une politique locale, et enfin d'analyser une politique ou de la comparer à des politiques construites par un humain, comme les politiques expertes.*

KEYWORDS: *Factored action factored Markov decision processes, policy optimization, policy analysis.*

MOTS-CLÉS : *Processus décisionnels de Markov à espace d'état et d'action factorisés, optimisation de politique, analyse de politique.*

1. Introduction

Systems management in ecology and agriculture is complex because several entities must be managed together and these entities are in strong interaction. Furthermore, management actions are applied at a local level while the objective is often defined at a larger scale. For instance, spatial dispersion of pests create spatial dependencies between fields which should be taken into account when designing management policies at the scale of an agricultural landscape. Another feature of these management problems is that the objective is a long-term one. For example, biodiversity and production have to be preserved in a sustainable way. Finally, sequences of decisions are taken without a precise and deterministic knowledge of the global delayed effects of the decisions.

Markov Decision Processes (MDPs) form a suitable tool for modelling and solving problems of sequential decision under uncertainty Puterman (1994); Sigaud, Buffet (2010). A MDP is defined in terms of state variables, action variables, transition probability functions and reward functions. Solving a MDP amounts to finding a policy that optimizes the expected sum of future rewards, over a given time horizon. There exists several freely available toolboxes for solving Markov Decision Processes[1].

However, the use of the MDP framework is not straightforward in the case of a large number of state variables: optimal policies cannot be computed or represented efficiently. SPUDD Hoey *et al.* (1999) and APRICODD St-Aubin *et al.* (2000)[2] approaches implement respectively exact and approximate solution approaches based on Algebraic Decision Diagrams. The aGrUM C++ library[3] also implement solution algorithms for factored MDP. A limit of these approaches is that they handle only a flat action space, while in the above mentionned applications, the action space is multidimensional. Several algorithms have been proposed for such MDPs (FA-FMDPs) with multidimensional state and action spaces Guestrin *et al.* (2001); Kim, Dean (2002). Most approaches for solving large FA-FMDPs compute approximate policies which are local, in the sense that the decision rule prescribes the action to apply in a particular entity based only on the current state of the few entities in direct interaction. One such approach is the Graph-based MDP (GMDP) framework Sabbadin *et al.* (2012); Cheng *et al.* (2013). In a GMDP, each entity is represented as the node of a graph. To each node is associated a pair of state / action variables. The graph edges represent local dependencies in the transition and reward functions. For a fixed policy, the dynamic model is a Dynamic Bayesian Network. Its graphical representation provides an easy interpretation of the local dependencies in the GMDP model.

1. *e.g.* MDPtoolbox: http://www.inra.fr/mia/T/MDPtoolbox Chadès *et al.* (2014),
Markov Decision Process Toolbox: http://www.cs.ubc.ca/∼murphyk/Software/MDP/mdp.html,
Markov Decision Process Toolbox for Python: https://pypi.python.org/pypi/pymdptoolbox.

2. https://cs.uwaterloo.ca/∼jhoey/research/spudd.

3. https://forge.lip6.fr/projects/agrum/wiki.

In this article, we introduce the Matlab GMDPtoolbox library, dedicated to GMDP. We recall the framework and we briefly describe and illustrate the main functionalities of the toolbox: representation of a structured decision problem in the GMDP framework, solution algorithms to finding good policies and finally tools to analyze a policy or compare it with human-built ones, like expert ones.

2. Problem representation

A discrete-time GMDP is defined by a 5-tuple $< S, A, N, p, r >$ where :

– S is the state space, $S = S_1 \times \cdots \times S_n$. S_i is the finite state space of site i.

– A is the action space, $A = A_1 \times \cdots \times A_n$. A_i is the finite action space of site i.

– N is the set of sites neighborhood relations, $N = \{N_i,\ \forall i = 1, \dots, n\}$ where $N_i \subseteq \{1, \dots, n\}$ is the set of neighbors of site i. Note that it is possible that $i \in N_i$, but this may not be the case.

– p is the set of local sites transition probability functions, $p = \{p_i(s'_i|s_{N_i}, a_i),\ \forall i = 1, \dots, n,\ \forall s'_i, s_{N_i}, a_i\}$, where $p_i(s'_i|s_{N_i}, a_i)$ is the (stationary) probability for site i of transitioning to s'_i at time $t+1$ given that at time t the neighborhood of the site is in state s_{N_i} and action a_i is performed.

– r is the set of local sites reward functions $r = \{r_i(s_{N_i}, a_i),\ \forall i = 1, \dots, n,\ \forall s_{N_i},\ \forall a_i\}$, with $r_i(s_{N_i}, a_i)$ the reward obtained from site i at time t when the neighborhood of site i is in state s_{N_i} and action a_i is performed.

In a GMDP transitions are *local* : if $s = (s_1 \dots s_n), s' = (s'_1 \dots s'_n)$ and $a = (a_1 \dots a_n)$ are the global state and action vectors, and denoting $s_I = \{s_i\}_{i \in I}, \forall I \subseteq \{1 \dots n\}$,

$$p(s'|s, a) = \prod_{i=1}^{n} p_i(s'_i|s_{N_i}, a_i), \forall s \in S, \forall s' \in S, a \in A$$

Rewards are also *local*, with respect to the same neighbourhood relation:

$$r(s, a) = \sum_{i=1}^{n} r_i(s_{N_i}, a_i), \forall s \in S, \forall a \in A.$$

GMDPtoolbox allows to represent any GMDP problem in a Matlab structure, whose attributes[4] represent the 5 elements defining a GMDP. GMDPtoolbox also provides a set of functions[5] that encode ready-to-use GMDP examples into this structure.

4. See the Reference manual: http://www.inra.fr/mia/T/GMDPtoolbox/ReferenceManual.pdf
5. See GMDPtoolbox list of functions: http://www.inra.fr/mia/T/GMDPtoolbox/documentation.html

3. Policy design

In a classical MDP, a function $\delta : S \to A$ assigning an action to each state is called a *stationary decision rule* or *policy*. Once a policy δ is fixed, it defines a stationary *Markov chain* over S, with transitions $p_\delta(s'|s) = p(s'|s, \delta(s))$. The problem of finding the optimal policy for a stationary MDP, or solving the MDP, is then

$$\text{Find } \delta^*, S \to A, \quad \text{s.t.} \quad v_{\delta^*}(s) \geq v_\delta(s), \forall s \in S, \forall \delta \in A^S \ .$$

Where $v_\delta(s)$ is the infinite horizon discounted value of a policy δ, applied to an MDP with initial state s, defined as: $v_\delta(s) = E\Big[\sum_{t=0}^{+\infty} \gamma^t r(s^t, \delta(s^t))|s^0 = s\Big], \forall s \in S$.

Because the GMDP framework is for larger problems than the ones tackled by MDP, only approximate solution policies are usually looked for, in GMDP problems. In a GMDP, the search space is limited to a subset of policies that exploit the notion of neighborhood, namely the set of *local policies*. A policy $\delta : S \to A$ is said to be *local* if and only if $\delta = (\delta_1, \ldots, \delta_n)$ where $\delta_i : S_{N_i} \to A_i$ (instead of $\delta_i : S \to A_i$).

Currently, two algorithms Sabbadin *et al.* (2012), providing local policies by approximate resolution of a GMDP, are implented in GMDPtoolbox. They usually find empirically good local policies Cheng *et al.* (2013). The first one, referred to as MF-API, exploits the structure of the neighborhood relations of the GMDP and computes a *mean-field approximation* of the value function of a policy. This algorithm belongs to the family of *Approximate Policy Iteration* (API) algorithms. The second one is a specific *Approximate Linear Programming* algorithm derived from the general class of ALP algorithms and adapted to the GMDP framework. We will referred to it as ALP. Previous experimental comparisons have shown that the two algorithms provide local policies of similar quality, outperforming naive policies such as greedy or random policies. However, the MF-API algorithm usually provides a higher-quality approximation of the expected value of the returned policy than the ALP algorithm, which is faster. Thus the two methods can be seen as complementary. We refer the reader to Sabbadin *et al.* (2012) for a full description of these two algorithms.

The function implementing the ALP algorithm uses one function of the Matlab Optimization toolbox. Furthermore the implemented algorithms are speeded up by using a function of the Matlab Parallel Computing toolbox.

4. Policy analysis

A given local GMDP policy is in general not easy to interpret or analyze. GMDP-toolbox provides a set of functions (see footnote 5) to help understanding what are the actions prescribed by the policy, site by site or globally, either in a synthetic format or graphically. From these statistics, it can be easier to extract a general decision rule from the local policy.

Furthermore, in order to evaluate the consequences on the rewards of applying a policy, the toolbox provides functions to compute the value of a policy, and to generate spatio-temporal simulations of the system under its application. This could also be useful to compare several policies, for example the one found by optimization with human-designed ones like experts ones. Note that some graphical representations rely on graphViz4Matlab toolbox[6].

5. Epidemics management toy example

In order to illustrate the use of the toolbox, a quick start guide[7] has been written to detail the main steps of a classical study on a toy epidemics management problem on 3 crop fields.

Each crop field can be in two states ($|S_i| = 2, \forall i = 1, 2, 3$): uninfected (coded 1) or infected (coded 2). Each field is susceptible to contamination by a pathogen. Contamination can either be long range, or over a short distance between neighboring fields. When a field is contaminated, the yield decreases. Decisions about crop field management are taken each year and two actions can be applied to each field ($|A_i| = 2, \forall i = 1, 2, 3$): a normal cultural mode is applied (coded 1) or the field is left fallow and treated (coded 2). The problem is to optimize the choice of a long-term policy in terms of expected yield. The topology of the area is represented by a graph (see Figure 1, left). There is one node per crop field. A directed edge between two nodes represents potential contamination. The neighborhood relationship is symmetric since we assume that infection can spread in any direction: $N_1 = \{1, 2\}, N_2 = \{1, 2, 3\}, N_3 = \{2, 3\}$.

Transition probabilities are parametrized by the following variables: the probability pd that a field infected becomes uninfected when it is left fallow and treated, the probability $p\epsilon$ of long-distance contamination, the probability pc that a field be contaminated by an infected neighboring field. The probability $p(m_i)$ that a field with m_i infected neighboring fields moves from state uninfected to infected under a normal cultural mode is then defined by $p(m_i) = p\epsilon + (1 - p\epsilon)(1 - (1 - pc)^{m_i})$.

Harvest is affected by the state of the crop field and the chosen action. The maximal yield (noted r) can be achieved for an uninfected field with normal treatment. If the field is contaminated, the reward is only $r/2$. Otherwise, a field left fallow and treated produces no reward. On Figure 1, right, is given the Dynamic Bayesian Network representation of this GMDP, illustrating interactions between state variables, action variables, and rewards.

On this toy example the MF-API and the ALP resolution algorithms lead to the same local policy (experiments were run with a discount factor equal to 0.95). This

6. graphViz4Matlab toolbox on MatlabCentral: http://www.mathworks.com/matlabcentral/fileexchange/21652-graphviz4matlab

7. GMDPtoolbox Quick start: http://www.inra.fr/mia/T/GMDPtoolbox/QuickStart.pdf

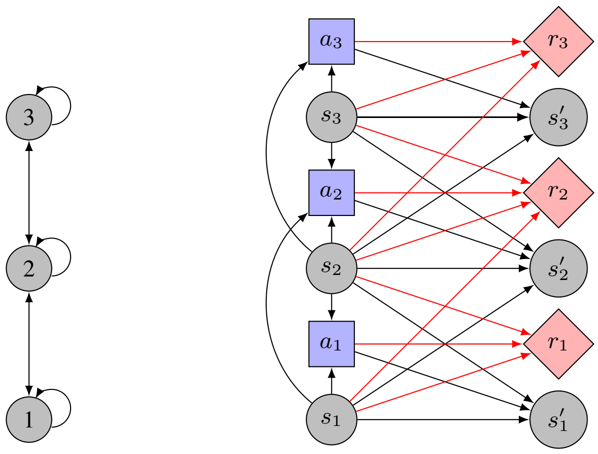

Figure 1. Epidemics management problem. Left: Graphical representation of the neighborhood relationships. Right: Corresponding Dynamic Bayesian Network (black edges) and reward function structure (red edges).

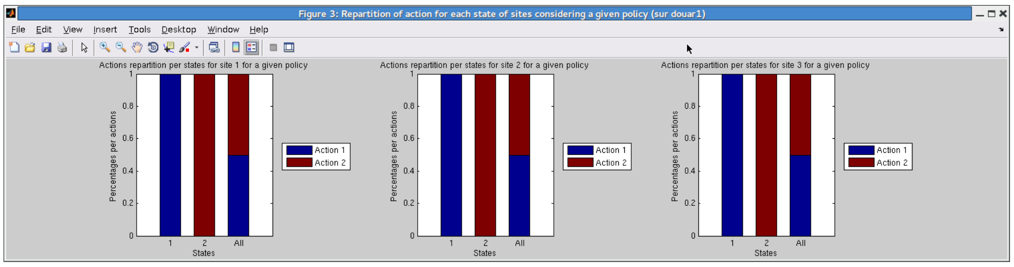
Figure 3: Repartition of action for each state of sites considering a given policy (sur douar1)
File Edit View Insert Tools Desktop Window Help
Actions repartition per states for site 1 for a given policy
Actions repartition per states for site 2 for a given policy
Actions repartition per states for site 3 for a given policy
Percentages per actions
States
1
2
All
Action 1
Action 2

Figure 2. Epidemics management problem: proportions of neighborhood configurations for which the GMDP policy prescribes action 1 and action 2, at each of the three sites.

policy can be difficult to interpret since it corresponds to a set of local functions δ_i from S_{N_i} to A_i. In GMDPtoolbox, one of the proposed visualizations allows to show the repartition of each action depending of the site state, and for each site (see Figure 2). From these graphics we can see that in this very simple example the policy amounts to the following simple rules, that depend only on the site status and not on the status of the neighboring fields: if field i is uninfected (site state 1) then choose a normal cultural mode (action 1); if field i is infected (site state 2) then leave the field fallow and treat (action 2).

Furthermore with GMDPtoolbox, it is possible to evaluate and plot the average value, over initial states, of the discounted finite horizon value of a policy, for increasing horizon (Figure 3, left). Evaluation is computed using Monte Carlo simulations. GMDPtoolbox also provides a graphical representation of the expectation of the sum of the discounted rewards at a given time step, over all sites (Figure 3, right).

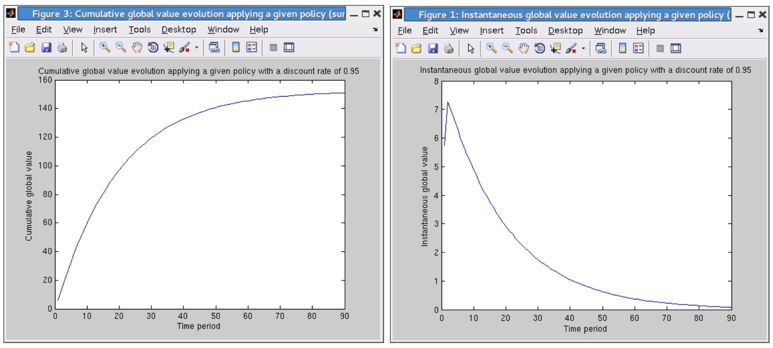

Figure 3. Epidemics management problem: expected cumulative and instantaneous global reward of the GMDP policy, estimated by Monte Carlo.

Other functions allow to quantify the contribution of each site to the cumulated global value, and the time spent by each site in the different states.

Conclusion

A complete description of GMDPtoolbox and the source code of the toolbox are available at http://www.inra.fr/mia/T/GMDPtoolbox and on MatlabCentral[8]. The toolbox provides a complete environment to easily perform first steps with the GMDP framework or to represent and solve a structured decision problem. A new version of GMDPtoolbox will soon be released to offer more analysis tools.

At a longer term, we intend to extend the toolbox with the resolution algorithm proposed in Cheng *et al.* (2013). This algorithm belongs to a family of approaches called Planning as Inference that exploit the proximity between Factored MDPs and DBNs. Kumar *et al.* (2011) has first established that the evaluation of stochastic policies in MDP amounts to computing marginal probabilities in a Bayesian Network. Cheng *et al.* (2013) have applied inference methods to solve GMDPs, and Radoszycki *et al.* (2015) have proposed a policy iteration algorithm based on DBN inference to solve general Factored MDPs. So, even if this is not the mainstream approach to solve factored MDPs, Bayesian Network inference approaches have already been successfully applied to approximately solve Factored MDPs, including GMDPs.

Regarding the application domain of the GMDP framework, it has already been used to model management problems and to derive policies in various fields: plant disease management Peyrard *et al.* (2007), human disease management Choisy *et al.* (2007), forest management Forsell *et al.* (2011), and invasive species control Nicol *et al.* (2015). Considering the progresses of graph modelling for complex problems,

8. GMDPtoolbox on MatlabCentral: http://www.mathworks.com/matlabcentral/fileexchange/49101-graph-based-markov-decision-processes–gmdp–toolbox

GMDPtoolbox could be a powerful tool to help to address complex problems of management in agriculture, epidemiology or ecology, among other domains.

Acknowledgements

This work has been partly supported by the European Union Seventh Framework Programme (FP7/ 2007-2013) under the grant agreement n265865- PURE.

References

Chadès I., Chapron G., Cros M.-J., Garcia F., Sabbadin R. (2014). MDPtoolbox: a multi - platform toolbox to solve stochastic dynamic programming problems. *Ecography*, Vol. 37, No. 9, pp. 916-920.

Cheng Q., Liu Q., Chen F., Ihler A. (2013). Variational planning for graph-based MDPs. In *Advances in Neural Information Processing Systems*, p. 2976-2984.

Choisy M., Peyrard N., Sabbadin R. (2007). A probabilistic decision framework to optimize the dynamics of a network evolution: application to the control of childhood diseases. In *ECCS: European Conference on Complex Systems.*

Forsell N., Wikström P., Garcia F., Sabbadin R., Blennow K., Eriksson L. (2011). Management of the risk of wind damage in forestry: a graph-based Markov decision process approach. *Annals of Operations Research*, Vol. 190, No. 1, pp. 57-74.

Guestrin C., Koller D., Parr R. (2001). Multiagent planning with factored MDPs. In *Proceedings of NIPS'01*, p. 1523-1530.

Hoey J., St-Aubin R., Hu A., Boutilier C. (1999). SPUDD: Stochastic planning using decision diagrams. In *Proceedings of UAI'99.* Stockholm, Sweden.

Kim K.-E., Dean T. (2002). Solving factored MDPs with large action space using algebraic decision diagrams. In *Proceedings of the 7th Pacific Rim International Conference on Artificial Intelligence*, p. 80-89.

Kumar A., Zilberstein S., Toussaint M. (2011). Scalable multiagent planning using probabilistic inference. In *Proceedings of the 22th international joint conference on artificial intelligence.*

Nicol S., Chadès I., Peyrard N., Sabbadin R. (2015). An optimal approach to managing two-species competition stopping the Gambusia fish invasion of Edgbaston Mound springs. In *ICCB: 27th International Congress for Conservation Biology.*

Peyrard N., Sabbadin R., Lo-Pelzer E., Aubertot J.-N. (2007). A graph-based Markov decision process framework applied to the optimization of strategies for integrated management of diseases. In *American phytopathological society and society of nematologist joint meeting.*

Puterman M. (1994). *Markov Decision Processes: discrete stochastic dynamic programming.* Wiley.

Radoszycki J., Peyrard N., Sabbadin R. (2015). Solving f3mdps: Collaborative multiagent markov decision processes with factored transitions, rewards and stochastic policies. In *Prima 2015: Principles and practice of multi-agent systems*, pp. 3–19. Springer International Publishing.

Sabbadin R., Peyrard N., Forsell N. (2012). A framework and a mean-field algorithm for the local control of spatial processes. *International Journal of Approximate Reasoning*, Vol. 53, No. 1, pp. 66–86.

Sigaud O., Buffet O. (Eds.). (2010). *Markov Decision Processes in Artificial Intelligence.* Wiley.

St-Aubin R., Hoey J., Hu A., Boutilier C. (2000). APRICODD: Approximate policy construction using decision diagrams. In *Proceedings of NIPS'00.* Denver, CO.