Proc. IROS-2011 Workshop on
Active Perception and Object Search in the Real World

Objects Search : a Constrained MDP approach

Matthieu Boussard and Jun Miura
Department of Computer Science and Engineering,
Toyohashi University of Technology

***Abstract*—We consider the object search problem, where a robot has to explore its environment in order to localize some objects. We use a two step process, where the robot first detects candidate objects, and later identifies them using another algorithm. Since there are several candidate objects and the outcomes of the object recognition algorithm are uncertain, we model the planning process as an MDP. Furthermore, we give a certain amount of time to the robot to fulfil its mission. This leads to a problem where we want the robot to find as many objects as possible and as fast as possible within a limited time. This paper shows early results for the deterministic case. We model this by a Constrained Deterministic MDP, and we propose an incremental algorithm, based on a sequence of Mixed Integer Linear Program to compute the policy.**

I. Introduction

In order to perform more and more complex tasks, robots have to get a better understanding of their environment. We are considering an indoor environment, like offices or personal houses, where the robot can interact with humans. One of the most important tasks for a mobile robot is to preserve its integrity, and thus has to know where it may go safely. This issue has been well studied and has leaded to many SLAM algorithms (Simultaneous Localization and Mapping) [1]. To fulfil complex tasks, the information present on this map is not sufficient anymore. Instead, the robot has to know what kind of objects are in its environment and should be able to locate them on a map. The problem of searching and locating those objects on the map is called the objects search problem.

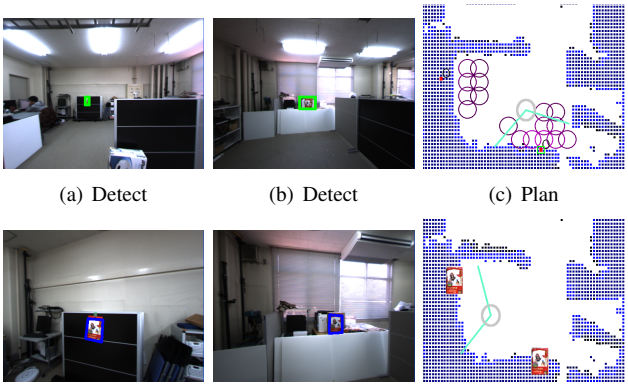
Many work has been done to search efficiently for objects. Sjöö et al. [2] present an attention mechanism and methods for depth computation, used to control the zoom level in order to perform an SIFT matching at an accurate distance measure. In [3], Meger et al. present *Curious George*, a combination between an attention system and a SLAM algorithm. This attention system allows the robot to take high definition pictures of potentially interesting area, which are used offline to perform the object detection. The principle of alternatively performing a move and an observation action is used by Shubina and Tsotsos in [4], where they compute the probability of an object's presence and the probability of an object detection using a certain type of recognition algorithm.

The lack of a long term policy, by selecting only the best next viewpoint, may lead to sub-optimal results. To obtain a long term plan, Aydemir et al. [5] are using a high level planner to select low level strategies to find a target object. The algorithm of Masuzawa et al. [6] that first detects candidates objects. Instead of directly selecting the best next viewpoint, they compute a long term policy. The authors rely on an ad-hoc world modelization in order to speed up their planning algorithm, but still, since they are performing an exhaustive search, is too slow to be solved for larger problems online.

Our problem is to recognize as many objects as possible and as fast as possible given a time limit. The contributions of this paper are the modelization of the problem as a Constrained Markov Decision Process (CMDP), a simplified Mixed Integer Linear Program (MILP) to solve it, and an incremental algorithm to control the calls to the MILP solver.

II. Problem

We define a candidate object as the location where something has been detected as potentially being a searched object. We call the location from where this candidate object can be identified a viewpoint. Fig.1 shows the object search problem. First, candidate objects are detected using a long range algorithm (here a color histogram search). Those detections are set as candidate objects on the map. Those candidates objects can be identified using an accurate and short ranged algorithm (here SIFT matches). For each object we define a set of viewpoints from where it is possible to apply the identification algorithm. We assume that this algorithm is not perfect and that we can estimate its probability of success for each viewpoint. We obtain a planning problem to select the optimal sequence of viewpoints, and once solved we can apply the selected action. At the end of the mission, we obtain a map augmented by objects information. In this paper, we will add a time constraint for the mission. We will focus on the planning algorithm, thus the exploration part will not be presented here. Furthermore, we will remove the uncertainty and will focus on how to manage the constrained planning problem.



(a) Detect (b) Detect (c) Plan

(d) Identify (e) Identify (f) Return map

Fig. 1. Object searching

III. MODEL

A. Markov Decision Processes

MDPs [7], [8] allow the formalization of a sequential decision problem under uncertainty. This process is fully observable, i.e. the observed state is the actual state of the system. A fully observable MDP is a 4-tuple $\langle S, A, P, R\rangle$

- S is the (finite) set of states,
- A is the (finite) set of actions,
- $P : S \times A \times S \rightarrow [0; 1]$ is the transition function,
- $R : S \times A \rightarrow \mathbb{R}$ is the reward function.

The unique optimal value function V^* is given by the Bellman equation [7] for the discounted expected reward for a discount factor $\gamma \in [0; 1]$. $\forall s \in S$:

$$V^*(s) = \min_{a \in A} \left(r(s, a) + \gamma \sum_{s' \in S} p(s, a, s')V^*(s') \right) \quad (1)$$

B. Observation planning model

From the raw sensor data, we need to build an MDP to compute the observation policy. We use model presented in [9], and defined by :

a) States set S: Since transitions are history independent, the states have to contain all previous information needed to take the decision. It is a way to handle the partial observability.

- the current position (x, y) of the robot,
- the list of visited viewpoints. Since the robot should not observe twice the same object from the same viewpoint, we need to keep the list of visited viewpoints $\{\{vp_1^1, vp_1^2, \ldots, vp_1^m\}, \ldots, \{vp_n^1, vp_n^2, \ldots, vp_n^m\}\}$ for all n objects, and vp_i^j is the jth observation for the ith object and m the maximum number of observations allowed.
- the information I_i about object i's status. it can be identified, rejected, or unknown.

A state s can be written :
$s = \langle x, y, \{\{vp_1^1, \ldots vp_1^m\}, \ldots, \{vp_n^1, \ldots vp_n^m\}\}, \{I_1, \ldots I_n\}\rangle$

b) Actions set A: The robot has only one kind of action. It is a macro action preforming a move action followed by an observation action. The robot selects a viewpoint, goes there and observes the related object. Its outcomes will be described by the transition function. We also add a *stop* action, available in every state, which makes the robot move to the starting position and reach the final state. This *stop* action is also executed when there is no more candidate objects.

c) Transition function P: In a general model, we consider move actions as deterministic and observation actions as stochastic. The probability of successfully identifying a candidate can be computed for each viewpoint according to known parameters (object type, location etc.). Since it is impossible to observe twice from the same viewpoint, this MDP is acyclic.

d) Reward and Cost function R: The observation planning problem can be naturally expressed by two criteria : the overall mission time, and the number of recognized objects. In [9], the authors focused on optimizing only the time criteria, whereas in this paper we want to recognize as many objects as possible given a time constraint. We define $C(s, a) < 0$ the cost (time) of executing a in s (the time to reach the viewpoint plus the time to recognize the object), and $R(s, a) > 0$ the reward for having identified an object (set to 1 if s has just recognized one object, 0 otherwise. This value can be changed to introduce preferences between objects).

In the following, we will restrict this general model to the deterministic case, where observations always succeed. This process becomes a Deterministic MDP. Observing one object immediately change its status to identified, hence the observation history can be removed from the state's definition. Even, this simplified model is a first step towards the stochastic one, the obtained plan can still be useful. For instance, it can be used to decompose the whole problem into the object observation order planning and the viewpoint planning for each object or as a heuristic value.

IV. CONSTRAINED MDP

As previously shown, the robot has to deal with rewards and costs of different natures and we don't optimize a weighted sum of the two criteria. Instead, we manage those two criteria separately through a CMDP, see [10] for a survey. It is an extension of MDP where the long term expected reward is subject to constraints on other resources. A mutli-criteria reinforcement learning algorithm has been proposed in [11] which ensures a minimum expected reward for every state before optimizing the other criterion. But it is working on a sub-class of CMDP where our problem can't be expressed.

When optimizing the number of objects using a time constraint, once satisfied, it is not optimized anymore. For instance, if a mission is given an infinite time, any policy that selects, for each object, the viewpoints having highest probability of recognition will be optimal, regardless to the global observation order! Even this behaviour is rational from an optimization perspective, it is unacceptable for a real application. Thus we have to optimize the mission's time using an expected number of recognized objects as a constraint.

The methods the most widely used are based on linear programming. The linear programming approach has been first introduced in [12]. We present here the dual of this linear program (LP) since it is more suited to solve CMDP [13]. The occupation measure $x_{s,a}$ represents the discounted number of time action a is taken in s, and $\alpha_s : S \rightarrow [0, 1]$ the initial probability distribution over states ; C being negative, the dual linear program to find the fastest policy is formulated as :

$$\begin{array}{ll} \text{Maximise} & \\ & \sum_{s,a} C(s, a)x_{s,a} \\ \text{Subject to} & \\ & \begin{array}{lcl} \sum_a x_{s',a} - \gamma \sum_{s,a} x_{s,a} p(s, a, s') & = & \alpha_{s'} \\ x_{s,a} & \geq & 0 \end{array} \end{array} \quad (2)$$

Once this linear program solved the optimal policy[1] can be computed by :

$$\pi(s,a) = \begin{cases} \frac{x_{s,a}}{\sum_a x_{s,a}}, & \text{if } \sum_a x_{s,a} > 0 \\ \text{arbitrary}, & \text{if } \sum_a x_{s,a} = 0 \end{cases} \quad (3)$$

It is possible to add extra constraints on the minimum expected number of recognized object R_{min} to the linear program LP.2. Those constraints are defined by the Eq.4 :

$$\sum_{s,a} R(s,a) x_{s,a} \geq R_{min} \quad (4)$$

Adding Eq.4 to LP.2 implies that the optimal policy becomes stochastic, which is not wanted. In [14] the authors showed that computing an optimal deterministic policy is NP-Complete. They compute a deterministic policy by adding a non linear constraint to the LP, $\forall s \in S, a, a' \in A, a \neq a'$:

$$|x_{s,a} - x_{s,a'}| = x_{s,a} + x_{s,a'}, \quad (5)$$

In [13], the authors change those additional constraints so that the mathematical program becomes an MILP. Since more tools are available to solve MILP than general mathematical program, the MILP may be easier to solve. They introduce $\Delta_{s,a}$ a binary variable to express the (unique) selected action a in s, and $X \geq x_{s,a}$ a constant to force $x_{s,a}/X \in [0;1[$. They compute the optimal deterministic policy by adding to LP.2 :

$$\begin{cases} \sum_a \Delta_{s,a} \leq 1 \\ x_{s,a}/X \leq \Delta_{s,a} \\ \Delta_{s,a} \in \{0;1\} \end{cases} \quad (6)$$

The CMDP defined to solve the observation planning problem, see Sec.III-B[2] has interesting properties : the starting state is known, it is acyclic and any policy will lead to the final state. Then we will use the same principle that combines LP.2 and Eq.6, but here, thanks to those properties, we can simplify Eq.6 by defining $x_{s,a}$ as binary variables and we finally propose the following MILP :

Maximise

$$\sum_{s,a} C(s,a) x_{s,a}$$

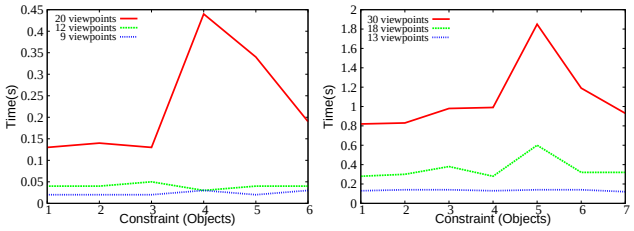
Subject to

$$\begin{aligned} \sum_a x_{s',a} - \sum_{s,a} x_{s,a} p(s,a,s') &= \alpha_{s'} \\ \sum_{s,a} R(s,a) x_{s,a} &\geq R_{min} \\ x_{s,a} &\in \{0;1\} \end{aligned} \quad (7)$$

Theorem 1: MILP.7 computes the optimal deterministic policy for an acyclic DMDP with unique and known starting state and $\gamma = 1$.

[1]Note that even the policy may appear stochastic, without constraint this policy is always deterministic

[2]We add for the constrained problem a stop action which can end the mission.



(a) 6 Candidate objects (b) 7 Candidate objects

Fig. 2. MILP solving time

Proof: The starting state s_0 is known, thus : for the starting state s_0, with $\gamma = 1$ and $\alpha_{s_0} = 1$ we have $\sum_a x_{s_0,a} = 1$. Since $x_{s,a} \in \{0;1\}$ one and only one action a_0 will be selected having $x_{s_0,a_0} = 1$. Since the problem is deterministic, there is only one state s' such that $p(s_0,a_0,s_1) = 1$, furthermore, since the MDP is acyclic, there is no other state s' such that $x_{s',a'} p(s',a',s_1) \neq 0$. We have $\sum_a x_{s',a} - x_{s_0,a} = 0$, and so on until the process reach the final absorbing state.■

V. Results

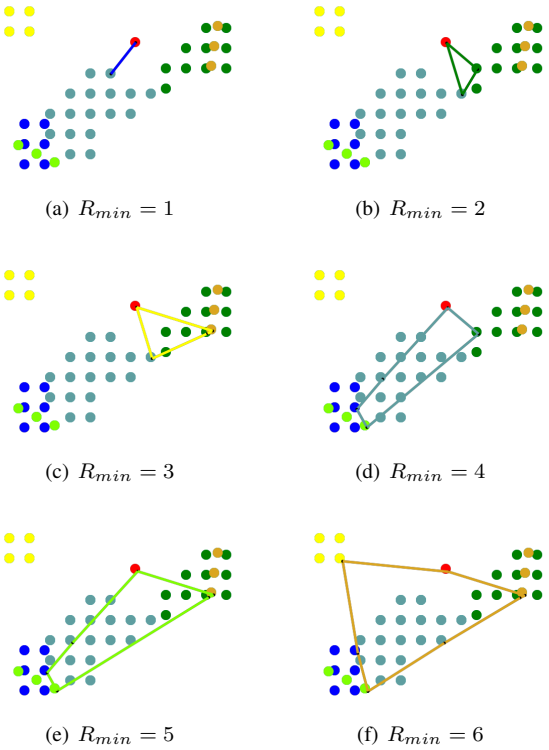
Fig.2 shows the computation time to solve the MILP for six and seven objects in the model, different number of viewpoints, and for various constraint value R_{min}. We use Ilog CPLEX with default options. For highly constrained problem ($R_{min} = 7$ obj) or low constrained problem ($R_{min} = 1$ obj), the optimal policy can be found quickly. But for "in-between" problems the computation time increases dramatically (6 candidates, 4 obj and 7candidates, 5 obj). Fig.3 shows the computed policy for different constraint values. In this picture, each color represents one object, and each circle represents one viewpoint for that particular object.

VI. Iterative MILP

We propose an iterative algorithm, Alg.1, which will at every step, find a solution to a problem constrained by a given minimum number of expected identified objects. Thus the solution found will be the fastest for that number of objects. We define $nbObj$ as the total number of candidate objects in the model. If the expected mission time $\sum_{s,a} C(s,a) x(s,a)$ is under the time limit, we increase the constraint value (line 6) in order to find a suitable plan. Fig.2 shows that some instances are very difficult to solve and should be avoided if possible. Alg.1 controls the search and can try to avoid those particular values when selecting R_{min} (line 6). For instance, if the robot has a lot of time, it can first plan for the maximum number of objects and, if succeed, doesn't need to solve for other values. When a little time remains, even many candidates could be checked, it is better to first search for a plan that recognize a few objects.

VII. Discussion

We showed how we can compute an observation plan for object recognition under time constraint. This is an early work, and the next step is to include uncertainty in the transition



(a) $R_{min} = 1$ (b) $R_{min} = 2$ (c) $R_{min} = 3$ (d) $R_{min} = 4$ (e) $R_{min} = 5$ (f) $R_{min} = 6$

Fig. 3. Policy execution for different constraint values R_{min}

function. The MILP can't simply use binary occupation measure variables anymore, we have to use one of the previous approach [13], [14]. Since the observations may fail, and since we want the robot to be able to try again, we have to keep a history of observation in the state, leading to a huge increment of the state space size. In previous works, this is solved by using Monte-Carlo algorithm and a limited horizon planning. It is possible to quickly compute a heuristic for the best policy to recognized all object (see [9]), and we can use it to get a high-level object recognition order, and then use it build an approximate, potentially sub-optimal, MILP. This observation order will force the plan to finish recognizing one object before continuing to the next one. In that case, the object observation history will be limited to the current object, the previous object being solved, and the next one not yet observed so the transition function will be limited to the current local plan.

Algorithm 1: Iterative MILP

Data: MDP $model$, mission time t_{max}
Result: π satisfying t_{max}
Generate MILP (see MILP.7) from $model$;
$R_{min} \leftarrow 1$;
repeat
  Set constraint R_{min};
  Solve MILP;
  $R_{min}++$;
until $E^{\pi}\left[\sum_{t=0}^{\infty}\gamma^t C_t | s_0 = s\right] > t_{max}$ *or* $R_{min} > nbObj$;
return π^*

VIII. Conclusion and Future Works

In this paper we presented the observation planning problem with limited time resource. We showed how we can use the properties of the observation planning problem to propose a simplified MILP. We showed early works using an iterative algorithm that solve a sequence of MILP. Once the observation planning problem is viewed as a MILP it is possible to use both the optimization techniques on the problem itself (Hierarchical planning, approximate MILP generation) or on the way of solving the MILP itself (approximate the solution of the generated MILP). Since the MILP are well-studied, having the observation planning expressed by those enables the use of many proved property, and also many efficient algorithms.

Acknowledgment

This work is supported by NEDO (New Energy and Industrial Technology Development Organization, Japan) Intelligent RT Software Project.

References

[1] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, September 2005.
[2] K. Sjöö, D. Gálvez-López, C. Paul, P. Jensfelt, and D. Kragic, "Object search and localization for an indoor mobile robot," *Journal of Computing and Information Technology, Special Issue on Advanced Mobile Robotics*, vol. 17, no. 1, 2009.
[3] D. Meger, M. Muja, S. Helmer, A. Gupta, C. Gamroth, T. Hoffman, M. Baumann, T. Southey, P. Fazli, W. Wohlkinger, P. Viswanathan, J. J. Little, D. G. Lowe, and J. Orwell, "Curious george: An integrated visual search platform," in *Proceedings of the 2010 Canadian Conference on Computer and Robot Vision*, ser. CRV '10. Washington, DC, USA: IEEE Computer Society, 2010, pp. 107–114.
[4] K. Shubina and J. K. Tsotsos, "Visual search for an object in a 3d environment using a mobile robot," *Comput. Vis. Image Underst.*, vol. 114, pp. 535–547, May 2010.
[5] A. Aydemir, K. Sjöö, J. Folkesson, A. Pronobis, and P. Jensfelt, "Search in the real world: Active visual object search based on spatial relations," in *Proceedings of the 2011 IEEE International Conference on Robotics and Automation (ICRA'11)*, Shanghai, China, May 2011.
[6] H. Masuzawa and J. Miura, "Observation planning for environment information summarization with deadlines," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Taipei, Taiwan, October 18-22 2010, pp. 30–36.
[7] R. Bellman, *Dynamic Programming*. Princeton University Press, 1957.
[8] M. L. Puterman, *Markov Decision Processes : Discrete Stochastic Dynamic Programming*. Wiley, 2005.
[9] M. Boussard and J. Miura, "Observation planning with on-line algorithms and GPU heuristic computation," in *ICAPS-10 Workshop on Planning and Scheduling Under Uncertainty*, May 2010.
[10] E. Altman, *Constrained Markov Decision Processes (Stochastic Modeling)*, 1st ed. Chapman & Hall/CRC, March 1999.
[11] Z. Gábor, Z. Kalmár, and C. Szepesvári, "Multi-criteria reinforcement learning," in *ICML*, J. W. Shavlik, Ed. Morgan Kaufmann, 1998, pp. 197–205.
[12] F. d'Epenoux, "A probabilistic production and inventory problem," *Management Science*, vol. 10, no. 1, pp. 98–108, October 1963.
[13] D. A. Dolgov and E. H. Durfee, "Stationary deterministic policies for constrained mdps with multiple rewards, costs, and discount factors," in *IJCAI*, L. P. Kaelbling and A. Saffiotti, Eds. Professional Book Center, 2005, pp. 1326–1331.
[14] E. A. Feinberg, "Constrained discounted markov decision processes and hamiltonian cycles," *Mathematics of Operations Research*, vol. 25, pp. 130–140, 1997.