# Optimal Control of Partiality Observable Markov Processes over a Finite Horizon

Report by Jalal Arabneydi

04/11/2012

Taken from “Control of Partiality Observable Markov Processes over a finite Horizon” by Richard D. Smallwood, Edward J. Sondik, 1973

## Abstract:

This report presents an approach to find exact solution of optimal control of POMDPs (Partiality Observable Markov Decision Process) over a finite horizon under having a few reasonable assumptions. The approach only considers finite-state Markov processes. By comparing MDPs and PODMPs from optimal control policies point of view, it will be demonstrated that solving POMDPs is harder than solving MDPs which highlights the importance of the approach. Next, by taking advantage of the fact that pay-off function is a piecewise-linear function, notion of α-vectors will be introduced. In addition, it will be shown α-vectors construct a structure for optimal control policy and there is no need to store every control action for every belief-state. In other words, α-vectors are completely sufficient to compute the optimal control action and it implies that the problem of optimal control of POMDPs is restricted to finding the α-vectors. Afterwards, an algorithm which calculates α-vectors will be presented and discussed in details. It will be proven that in the algorithm, number of points required is equal to the number of α-vectors. At the end, key points and features of the approach are summarized and a conclusion is made.

## Introduction:

A Partially Observable Markov Decision Process (POMDP) is a generalization of a Markov Process Decision. A POMDP models an agent decision process in which it is assumed that the system dynamics are determined by an MDP, but the agent cannot directly observe the underlying state. Instead, it must maintain a probability distribution over the set of possible states, based on a set of observations and observation probabilities, and the underlying MDP [4].

POMPDs have a very wide range of applications and some of them are mentioned to illustrate the wide-spread variety of usage of POMDPs. Eckles[1] has considered the control of POMDPS as applied to machine replacement problem. Smallwood has utilized POMDPs in human learning process [2]. Decoding of Markov sources transmitting over a noisy channel is another application of POMDPs, by Drake [3].

## MDPs and POMDPs from optimal control policy point of view:

This part seeks reasons of which make solving POMDPs more difficult than solving MDPs.

**MDPs:**

Consider MDP case and let X be a finite-sate space with Markovian properties:

Fig 1) $x \epsilon X = \{1,2, \ldots., n\}$

X is MDP, hence $p(x_{t+1}|x_t, u_t) = p(x_{t+1}|x_{1:t}, u_{1:t})$ . By having some properties on instantaneous cost and being independent of control polies for cost and transition probability, dynamic programming framework can be used for solving optimal control policy. Therefore, value function has following structure:

$$V_T(x_T) = \max_{u_T}(R(x_T, u_T)) \quad \textbf{(1)}$$

$$\vdots$$

$$V_t(x_t) = \max_{u_t}(R(x_t, u_t) + E(V_{t+1}(x_{t+1})|x_t, u_t))$$

Notice that for every state, one value function is defined and correspondingly there is one optimal control associated with that value function and state. Since the number of states is finite, it implies that the number of value functions at each time-step is finite. So, by doing the recursion, one can compute value functions at each time-step. Every optimal control should be stored and they form a table which can be expressed by a matrix whose raw represents time-step and its columns contain optimal control for one particular state over time-steps:

$$\begin{bmatrix} U_T(1) & U_T(2) & \cdots & U_T(n) \\ \vdots & & \ddots & \vdots \\ U_t(1) & U_t(2) & \cdots & U_t(n) \end{bmatrix}$$

Optimal control for time t, is (T-t+1)th raw of the matrix $[u_t(1) \;\; u_t(2) \ldots u_t(n-1) \;\; u_t(n)]$ . As a result, in MDP case, number of value functions at each step is finite due to finite number of states (finite-state Markov processes) and optimal control policies form a table.

**POMDPs:**

In the real world, it is usually not possible to access exact values of all states which implies the previous method, which is based on knowing the values of states exactly, cannot be used. Nevertheless, by doing some measurements, observations are accessible which are noisy versions of states, $y_t = h_t(x_t, n_t)$.

It has been proven in the literature that it is possible to tackle this problem in a similar manner as MDP is solved by defining belief state as follows: $\pi_t = p(x_t | y_{1:t}, u_{1:t-1})$.

It has also been shown that this belief state is MDP ($p(\pi_{t+1}|\pi_t, u_t) = p(\pi_{t+1}|\pi_{1:t}, u_{1:t})$ ), and under having some properties dynamic programming can be employed in this problem, similar to MDP cases with a difference that states in POMDPs are belief states.

$$V_T(\pi_T) = \ max_{u_T}\left(E\big(R(x_T, u_T)\big)\right)$$

$$V_t(\pi_t) = \ max_{u_t}\{\textstyle\sum_{i=1}^{n} R(x_i, u_t)\pi_i + E(V_{t+1}(\pi_{t+1})|\pi_t, u_t)\}$$

Notice that for every belief state, one value function is defined and one optimal control associated with that belief state and value function. Since belief state is continuous, unlike the states in MDPs, there is unaccountably infinite number of value functions at each time-step. Hence, it is impossible to do traditional DP, equation (1), due to infinite number of value functions at each time-step. To illustrate the continuity of belief state, two examples are given:

Example 1) Two-state case:

$$x_t \epsilon \{1,2\}, \quad p_t \triangleq P(x_t = 1) \rightarrow \pi_t = [p_t \quad 1 - p_t]$$

It is obvious that belief space is a line and has only one dimension as shown in Fig(2) that any point between [0,1] represents a probability distribution over state space. For instance, $\pi_t = [0 \quad 1]$ means that at time t with probability zero, current state is state 0 and with probability one it is state 1.

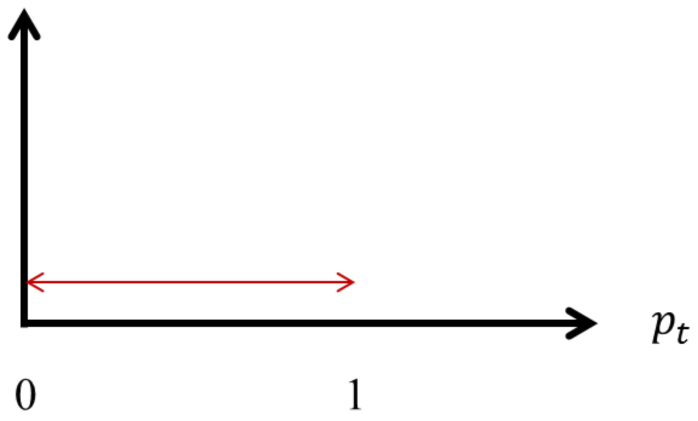


Fig 2) belief space is a line in two-state case

Example 2) Three-state case:

Belief space will be an equilateral triangle in $\mathbb{R}^2$.

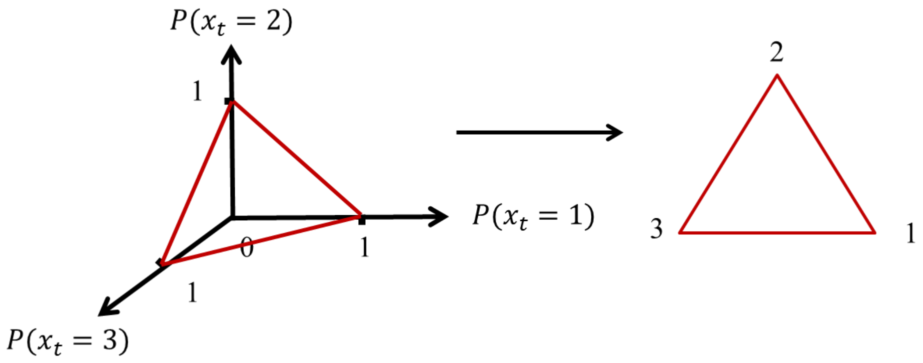


Fig 3) belief space is an equilateral triangle in three-state case

In general, n-state case has n states meaning state space is a subspace in $\mathbb{R}^n$and because there is only one constraint on summation of all probabilities to be 1, belief space belongs to $\mathbb{R}^{n-1}$ :

$$\sum_{i=1}^{n} \pi_i = 1 \xrightarrow{\text{yields}} \pi \epsilon \mathbb{R}^{n-1}$$

To sum up this part, it has been seen that belief space is continuous unlike the state space in MDPs. This continuity makes solving POMDPs harder than solving MDPs. One way to deal with this continuity is to quantize the belief space to a large number of states. In other words, the problem is converted approximately from continuous space to a discrete space and then it can be treated and solved by traditional techniques for MDPs, as developed by Howard[2]. However, once belief space is big, it requires a tremendously large number of quantization which makes this method not practical.

For example, if a quantization interval of 0.05 is used, then a five-state process will require 6.2 million states in the quantized process and this is completely impractical. Below, a table comparing MDPs and POMDPs in terms of number of value functions and structure of value functions is shown.

| | MDP | POMDP |
|---|---|---|
| # of value function | Finite | Infinite |
| Structure of optimal control policy | Table | $\alpha$-vector |

Table 1)

## **Introducing α-vectors** :

So far, it has been shown that solving POMDPs is hard due to unaccountably infinite number of belief states. To tackle the problem at first step it will be proven not rigorously, that value function is a piecewise linear function in current belief state. Then, by using this fact, concept of α-vectors is introduced. Next, under having a few assumptions, it will be proved that α-vectors partition belief space to a finite number of regions. In addition, optimal control can be completely achieved by only knowing a finite number of α-vectors. In other words, the problem of optimal control for POMDPs is limited to fining α-vectors who completely characterize the optimal control policy.

It is necessary to have following three assumptions to be able to show that the number of α-vectors is finite.

**Assumptions:**

1) Finite horizon
2) Finite set of observation
3) Finite set of control actions

$u_t \epsilon U = \{u_1, u_2, \dots, u_m\}$ , $\pi_t \epsilon \mathbb{R}^s$, $y_t \epsilon Y = \{y_1, y_2, \dots, y_q\}$

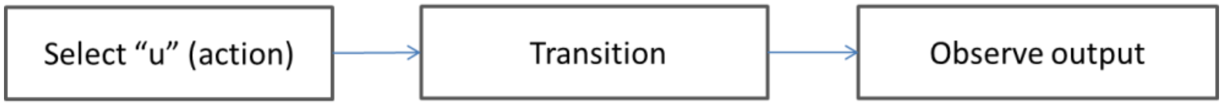


Fig 4) Sequence of operations

**Claim:** Value function is piecewise-linear function and convex (concave) due to maximizing reward (minimizing cost). Also, for each of these linear pieces (called α-vectors), there is one control action associated with it. In other words, knowing α-vector will result in knowing the control action which has generated the α-vector. Notice that knowing control action does not necessarily result in specifying associated α-vector because there might be many α-vectors generated by one control action.

As usual, starting step is calculating the value function at last step. There will be no future cost at last step, so it yields:

$$V_T(\pi_T) = \max_{u_T}\left(E\left(R(x_T, u_T)\right)\right) = \max_{u_T \epsilon U}( E\left(R(x_T, u_1)\right), E\left(R(x_T, u_2)\right), \dots, E(R(x_T, u_m)))$$

$$= \max_{u_T}\left\{\sum_{i=1}^{s} R(x_i, u_1)\pi_{i,T}, \sum_{i=1}^{s} R(x_i, u_2)\pi_{i,T}, \dots, \sum_{i=1}^{s} R(x_i, u_m)\pi_{i,T}\right\}$$

R is instantaneous reward. This expression can be re-written in vector-form:

$$\pi_T = \begin{bmatrix} \pi_{1,T} \\ \pi_{2,T} \\ \vdots \\ \pi_{n,T} \end{bmatrix} \rightarrow V_T(\pi_T) = \max_u\{\alpha(u_1)\pi_T, \alpha(u_2)\pi_T, \dots, \alpha(u_3)\pi_T\} \qquad \mathbf{(2)}$$

It is seen that value function at last step satisfies the claim. It is obvious that $V_T$ is a piecewise linear function in current belief state and each linear piece (α-vector) is completely addressed by one control action. Since action space is finite then the number of these α-vectors is finite. So, α-vectors are sufficient to compute the optimal control action and there is no need to store control action for every belief state. Intuitively, one can observe a mapping from infinite space to a finite number of classes $\pi \rightarrow \alpha - \text{vectors}$ and that is the trick used to handle the continuity problem of belief state. Fig 5) shows a typical form of value function in two-state case:

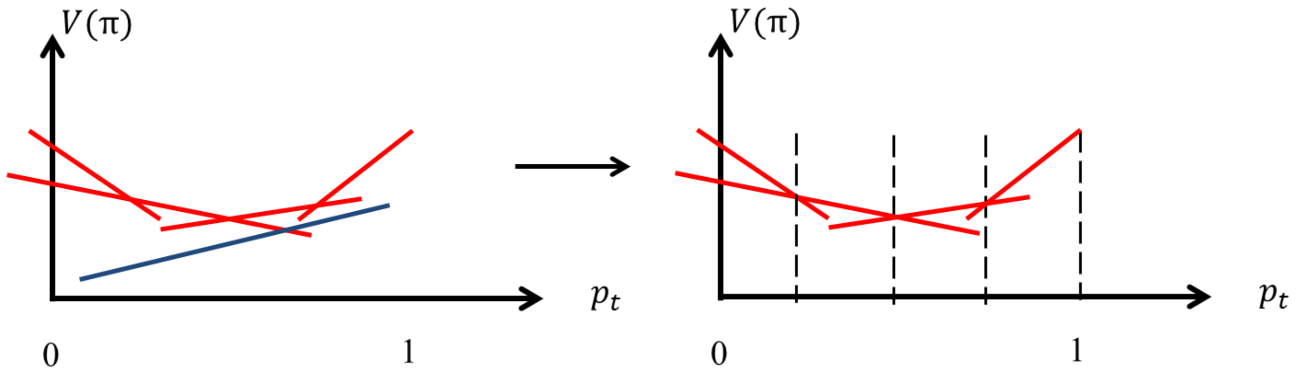


Fig 5) There may exist α-vectors who get completely dominated by other α-vectors, so they have no impact on optimal control policies (left hand side picture). α-vectors partition the belief space [0,1] (right hand side picture). Notice that within each partition, α-vector is fixed and it is critical to understand that each partition (region) can be characterized by its corresponding α-vector.

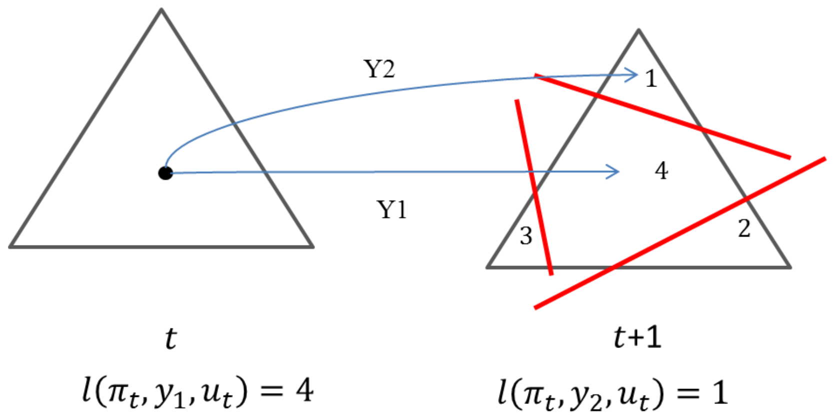
Assume that all the α-vectors for time step t+1 are known:

$$V_{t+1}(\pi_{t+1}) = \max_{u_{t+1}} \ (\alpha_{t+1}(u_{t+1}).\pi_{t+1})$$

Each α-vector is addressed by index L which is a function of $\pi_{t+1}$. Knowing $\pi_{t+1}$ results in knowing what the corresponding α-vector is. To compute α-vector given$\pi_{t+1}$, one needs to plug in $\pi_{t+1}$ to all α-vectors and observes which α-vector gives the highest value. Whoever gives the largest is the α-vector which partitions the belief space to a region at which $\pi_{t+1}$ belongs to. Hence:

$$L(\pi_{t+1}) \triangleq 1,2,\dots,\Omega = (\#\text{ of all }\alpha - \text{vectors at step t} + 1\text{ i.e. }\#\text{ of all partitions(regions)})$$

On the other hand, $\pi_{t+1} = T(\pi_t, y_{t+1}, u_t)$ due to POMDPs property. Finally, function $l$ is defined such that $l(\pi_t, y_{t+1}, u_t) \triangleq L(\pi_{t+1}) \triangleq 1,2,\dots,\Omega$ .



$$\pi_{t+1} = T(\pi_t, y_{t+1}, u_t)$$

Fig 6) If current belief state is $\pi_t$ and action $u_t$ is applied and $y_1$ is observed, next belief state lies inside region 4 which means the value of $l(\pi_t, y_1, u_t) = 4$.Under the same assumption and if $y_2$ is observed, next belief state goes to region 1 and $l(\pi_t, y_2, u_t) = 1$, respectively.

According to right hand side picture of Fig (6), there only 4 α-vectors and over each region the corresponding α-vector is the same. Each region is addressed by a number $l(\pi_t, y_{t+1}, u_t)$ and it is the index of the region.

$V_{t+1}$ is piecewise linear and convex (convexity is obtained due to the fact that maximum over a set of piecewise linear convex functions is itself piecewise linear and convex). As illustrated in the Fig (6), notice that for each pair of action and observation, $l(\pi_t, y_{t+1}, u_t)$ is a finitely valued function of $\pi_t$. This fact plus the assumed convexity of $V_{t+1}$ and continuity of $\pi_{t+1} = T(\pi_t, y_{t+1}, u_t)$ Imply that $l(\pi_t, y_{t+1}, u_t)$ partitions the belief space into a finite number of regions such that $l(\pi_t, y_{t+1}, u_t)$ is single-valued over each region.

Now, our claim should be true for time-step t:

$$V_t(\pi_t) = \max_{u_t}\{\sum_{i=1}^{n} R(x_i, u_t)\pi_i + E(V_{t+1}(\pi_{t+1})|\pi_t, u_t)\} \qquad \mathbf{(3)}$$

$$\pi_{t+1}(x_{t+1}) = \frac{p(y_{t+1}|x_{t+1}).p(x_{t+1}|\pi_t,u_t)}{p(y_{t+1}|\pi_t,u_t)} \qquad \mathbf{(4)}$$

By substituting equation (4) in the second term of bracket in equation (3), it yields:

$$E(V_{t+1}(\pi_{t+1})|\pi_t,u_t) = \sum_{y_{t+1}} p(y_{t+1}|\pi_t,u_t).V_{t+1}\left(\begin{bmatrix} \frac{p(y_{t+1}|x_{t+1}=1).p(x_{t+1}=1|\pi_t,u_t)}{p(y_{t+1}|\pi_t,u_t)} \\ \vdots \\ \vdots \\ \frac{p(y_{t+1}|x_{t+1}=s).p(x_{t+1}=s|\pi_t,u_t)}{p(y_{t+1}|\pi_t,u_t)} \end{bmatrix}\right)$$

$$= \sum_{y_{t+1}} V_{t+1}\left(\begin{bmatrix} p(y_{t+1}|x_{t+1}=1).p(x_{t+1}=1|\pi_t,u_t) \\ \vdots \\ p(y_{t+1}|x_{t+1}=s).p(x_{t+1}=s|\pi_t,u_t) \end{bmatrix}\right) \qquad \mathbf{(5)}$$

Since $V_{t+1}$ is piece-wise linear, the effect $p(y_{t+1}|\pi_t,u_t)$ will cancel out. Also, note that the effect of observation on problem is just $p(y_{t+1}|x_{t+1})$ which only depends on $(y_t = h_t(x_t,n_t))$ and it is independent of $\pi_t, u_t$. So, this summation is nothing but a weighted average.

$$p(x_{t+1}|\pi_t,u_t) = \sum_{x_t} p(x_{t+1}|x_t,u_t).\pi_t \qquad \mathbf{(6)}$$

By substituting equation (6) in (5):

$$\sum_{y_{t+1}} V_{t+1}\left(\begin{bmatrix} p(y_{t+1}|x_{t+1}=1).\sum_{x_t} p(x_{t+1}=1|x_t,u_t).\pi_t(x_t) \\ \vdots \\ p(y_{t+1}|x_{t+1}=s).\sum_{x_t} p(x_{t+1}=s|x_t,u_t).\pi_t(x_t) \end{bmatrix}\right) =$$

$$\sum_i \alpha'_{i,t}(u_t).\pi_{i,t} = \alpha'_t(u_t).\pi_t \qquad \mathbf{(7)}$$

(7) is clearly a piece-wise linear function in $\pi_t$ and also the number of resulting α-vectors is finite. Moreover, for each α-vector, there is a control action associated with it

Equation (3) can be re-written as maximization over m different possible choice of actions. Now, let us take a look at the value associated to each control action (u):

$$r_u.\pi_t + \sum_y V_{t+1}(\pi_{t+1})|\pi_t, u, y \qquad \mathbf{(8)}$$

Notice that for each term, control action is fixed. If the summation over y is expanded, one can see each term of it is a weighted value of $V_{t+1}(\pi_{t+1})$ given $(\pi_t, u, y)$. On the other hand, knowing $(\pi_t, u, y)$ is sufficient to indicate index $l(\pi_t, y, u)$ which specifies in what region next belief state lies. It is equivalent to know what ${\alpha_{t+1}}^{l(\pi_t, y, u)}$ models $V_{t+1}(\pi_{t+1})$ , exactly, at $\pi_{t+1}$ given action u and observation y. Summation over y has q terms, each of which has different observation and it implies that for each of them $l(\pi_t, y, u)$ can be different. As a result, equation (8) depends on $l(\pi_t, y, u)$. From equation 6, it is obvious, equation (8) is linear in $\pi_t$ and a function of control action. Therefore, associated expected reward for control action $u$ can be written as follows:

$$\alpha^{l(\pi,y,u)}{}_t(u).\pi_t \triangleq r_u.\pi_t + \sum_y V_{t+1}(\pi_{t+1})|\pi_t, u, y$$

So:

$$V_t(\pi_t) = \ max_{u_t}\{\alpha^{l(\pi_t,y,u_t)}{}_t(u).\pi_t\} \qquad \mathbf{(9)}$$

In worst case scenario, equation (8) may have Ω (size of all α-vectors in time t+1) different representations (α-vectors). By changing$\pi_t$, $l(\pi_t, y, u)$ may change and in the worst case, it travels all the regions available in time t+1which means all of α-vectors in step t+1. So equation (8) takes Ω different linear models in worst case i.e. it creates Ω different α-vectors. Since maximization is over m actions, $m.\Omega$ α-vectors will be generated at time t. Knowing that at t=T, $\Omega =$ m, results in $m^n$ α-vectors, (n is control horizon). Size of α-vectors grows exponentially in control horizon and that is why assumption of finite horizon is needed. To sum up this part:

**1)** α-vector associated with any belief state ($\pi_t$) can be computed by knowing all α-vectors of $V_{t+1}$. In order to do this, following steps need to be taken given $\pi_t$ :

① Construct m equations of form (8), for each control action.

② Number of actions and observations is finite, so for each pair of action and observation given $\pi_t$ , $l(\pi_t, y, u)$ is known. Therefore, model of $V_{t+1}$ will be known and all possible a-vectors can be computed then.

③ Plug $\pi_t$ in all of computed a-vectors, whoever gives the highest value is the a-vector associated with $\pi_t$.

**2)** α-vectors are sufficient for knowing optimal control.

Hence, problem of finding optimal control policy for POMDPs is restricted to calculating α-vectors. On the other hand, α-vectors for the last step are already known based on equation (2), so if an algorithm can generate new α-vectors from previous (old) α-vectors, the optimal control problem is solved.

## How to generate new α-vectors given old (one-step back) α-vectors:

Given a current belief state and old α-vectors, it was shown corresponding α-vectors can be completely computed in previous part. One simple approach to find new α-vectors is to plug in a large number of belief states and compute α-vectors for each of them with the hope that all possible α-vectors will be detected by this large set of belief states. However, there is no guarantee that all possible α-vectors will be discovered using this approach.

Proposed algorithm by Smallwood and Sondib guarantees to find all new α-vectors and number of points chosen equals the number of all new α-vectors.

The basis of the algorithm is on picking points wisely instead of choosing them randomly. So, points should be picked such that each of them results in a new α-vector. In other words, algorithm is based on finding a next point such that its corresponding α-vector is new and not being achieved so far. To do this, a point at belief space is picked randomly and then its corresponding α-vector and control action will be calculated, (call them $,\alpha^*, u^*)$ . Now, for the next point, it is necessary to choose a point such that it does not belong to a region over which all belief states result in $\alpha^*$. In other words, if the region associated with $\alpha^*$ can be computed somehow, then any point outside of this region can be chosen as the next point and it certainly results in a different α-vector .Therefore, the region over which $\alpha^*$ is fixed needs to be computed.

$$\pi = current\ belief\ state\ , \qquad \alpha^* = \alpha - vector\ associated\ with\ \pi$$

$$u^* = optimal\ control\ action\ associaten\ with\ \alpha^*$$

In order to specify the region, we move slowly away from $\pi$ and check when we enter another region i.e. when α-vector changes. According to equation (9), α-vector changes only if the argument of maximization changes by moving slowly away from $\pi$ to $\pi_t{}'$. The argument in equation (9) is only a function of $l$ and $u$. So, a change in the argument will be made in two ways, it is either amount of $l(\pi, y, u)$ changes while $u$ is fixed or $\alpha^l{}_t(u)$ changes for different actions. Therefore, for computing the region, we move away slowly form $\pi$ and then see for which $\pi_t{}'$, we will still be in the region and once $\pi_t{}'$ enters another region, it implies that $\pi_t{}'$ is on the boundary.

**Type-1 condition:**

This condition refers to the scenario once L changes while control action is fixed. In this case, due to the fact $l$ changes, it implies that we will have a change in regions in step t+1 which means $V_{t+1}(\pi_{t+1}) = \max_{u_{t+1}} \ (\alpha_{t+1}(u_{t+1}).\pi_{t+1})$ should change the α-vector. Thus, In order to still stay in the region associated with $\alpha^*$ at $\pi_t{}'$, we need the following inequalities to be hold:

$$\alpha_{t+1}{}^{l(\pi_{t}{}',u^*,y)}\pi'{}_{t+1} \geq \alpha_{t+1}{}^{k}\pi'{}_{t+1} \qquad for\ all\ k$$

$$\pi'{}_{t+1} = T(\pi_t{}', u, y)$$

$$\alpha_{t+1}{}^{l(\pi_{t}{}',u_t{}^*,y)}T(\pi_t{}', u_t{}^*, y) \geq \alpha_{t+1}{}^{k}\ T(\pi_t{}', u_t{}^*, y) \quad for\ all\ k, for\ all\ y \qquad (\mathbf{10})$$

These inequalities should be hold for every possible observation, so there will be a huge number of inequalities. Fortunately, most of these inequalities can be discarded and only those inequalities whose values of k is such that the region associated with $\alpha_{t+1}{}^{k}$ forms a boundary with the region for $\alpha_{t+1}{}^{l}$ . Notice that in this type, even though a new α-vector is achieved, optimal control still remains the same, $u^*$.

**Type-2 condition:**

This condition occurs once there is a change in α-vector for $V_t(\pi_t{}')$. By taking a look at equation (9), we see that in order to stay in the region associated with $\alpha^*$ at $\pi_t{}'$ , following inequalities required to be hold:

$$\alpha^*{}_t.\pi'{}_t \geq \alpha_t(u).\pi'{}_t \qquad for\ all\ control\ actions \qquad (\mathbf{11})$$

In this type, by going to a different region, a new α-vector is obtained. Also, control action will be changed and will not be $u^*$.

Apart from these two types of conditions, there are other inequalities needed to be hold too::

$$\pi'_i \geq 0 \ , \quad \sum_{i=1}^{s} \pi'_i = 1 \qquad (\mathbf{12})$$

All these inequalities together will specify the region of belief space over which α-vector $\alpha^*$ defines the optimal value function and of course the control action $u^*$ is the optimal in this region. Generally only some subsets of these linear constraints are necessary to define the region that is to say, some subset of these hyper-planes will not be the boundaries for the region. Employing linear programming algorithm enables us to identify constraints defining the region. Applying linear programming to the linear inequalities in (10), (11), and (12) achieves a minimum subset of inequalities that define the region.

Now, we can pick a point outside of the region which results in a new α-vector and depending on either it is of type 1 or 2, the control action will remain the same or will change respectively. We keep doing this procedure to find all the possible new α-vectors.

**Algorithm:**

**1)** pick a belief state randomly and then compute its corresponding α-vector and optimal control action $(\alpha^*, u^*)$

**2)** Set up all the following inequalities in order to find the region over which α-vector is $\alpha^*$.

$$\alpha_{t+1}{}^{l(\pi_t{}', u_t{}^*, y)} T(\pi_t{}', u_t{}^*, y) \geq \alpha_{t+1}{}^{k}\, T(\pi_t{}', u_t{}^*, y) \qquad \mathit{for\ all\ k\ and\ y}$$

$$\alpha^*{}_t . \pi'_t \geq \alpha_t(u). \pi'_t \qquad \mathit{for\ all\ control\ actions}$$

$$\pi'_i \geq 0 \ , \quad \sum_{i=1}^{s} \pi'_i = 1$$

3) Utilize linear programming to find the minimum set of inequalities defining the region. From each boundary of the region, a new α-vector is computed and then store it along with its associated optimal control and one belief state for which it is α-vector.

4) Store the indices of α-vectors that are neighbors to the region under consideration to limit the number of inequalities of type 1 during the running this algorithm for time (t-1). If there are any α-vectors on the list whose region has not been calculated, choose a new α-vector and go to step 2. Otherwise, the complete specification of the optimal control policy has been calculated.

**Linear Programming:**

All inequalities in (10) and (11) can be written in the following format:

$$\pi . b^z \geq 0 \quad where\ z\ varies\ over\ all\ constarints\ in\ (10)\&(11)$$

Thus, solution to linear program

$$\min_{\pi} \pi . b^k \qquad\qquad z = 1,2,3, \ldots \qquad (A)$$

Subject to $\pi_i \geq 0$ , $\sum_{i=1}^{s} \pi_i = 1$

Will achieve a solution that has the slack variable for the kth inequality equal to zero if and only if this inequality forms a part of boundary of the region. Hence, by solving a linear program of the form in A, for each of the constraint, we can identify the constraints that define the region and the ones that can be discarded.

## Conclusion:

At the first stage, the importance of this approach was demonstrated by doing a comparison between MDPs and POMDPs and it was shown that traditional approaches for MDPs will fail to work for POMPDs in general cases, due to continuity nature of belief state. At the second stage, based on the fact that value function is a piecewise linear function in belief state, concept of α-vectors were introduced. It was proven that optimal control for each belief state is completely characterized by α-vectors and there is no need to store every optimal control action for every belief state. Moreover, α-vectors partition the belief space and in the worst case scenario, the number of α-vectors grows exponentially as control horizon increases $m^n$, where m is the size of action space and n is control horizon. At the third stage, it was shown that problem of optimal control for POMDPs is limited to fining α-vectors. Consequently, an algorithm was presented which guarantees to find all α-vectors and only requires as same number of points as the number of α-vectors.

It is trivial to consider a discount factor on the second term of value function. In MDPs, optimal control policy is a mapping from state space to action space: $U_t = g(x_t) \quad , \quad x \ \rightarrow u.$ However, in POMDPs, it is expected to have a mapping from belief space to action space for optimal control problems qualified to use dynamic programming, and one can think of α-vectors as an intermediate mapping which connects belief space to action space.

$$U_t = g(\pi_t) \qquad\qquad \pi \xrightarrow{using\ algorithm} \alpha - vectors \xrightarrow{Max} u$$

## References:

**1)** James E. Eckles, “Optimum Maintenance with Incomplete Information”, Opns. Res.16, 1058-1067, (1968).

**2)** Richard D. Smallwood, I. Weinstwen, and J. Echles, “Quantitative Methods in Computer-directed Teaching systems”, Final Report Nonr-225(84), Department of Engineering-Economic Systems, Stanford University, Stanford, California, March 15, (1967).

**3)** Alvin Drake, “Observation of a Markov Process through a Noisy Channel”, Sc.D. Thesis, Electrical Engineering Department, Massachusetts Institute of technology, Cambridge, Massachusetts, (June 1962).

**4)** http://en.wikipedia.org/wiki/Partially_observable_Markov_decision_process

**5)** Richard D. Smallwood, Edward J. Sondik, “The Optimal Control of Partially Observable Markov Processes Over a Finite Horizon”, Operations research, Vol. 21, No. 5, 1071-1088, (Sept.-Oct., 1973).