Constrained Multi-object Markov Decision Scheduling with Application to Radar Resource Management

Mohammad Rezaeian and Bill Moran
Department of Electrical and Electronic Engineering,
University of Melbourne, Australia
rezaeian, b.moran @ee.unimelb.edu.au

Abstract – *Hierarchial radar resource management uses multi object Markov decision scheduling with a constraint on the resources. In this paper we give a detailed description of constrained multi-object Markov decision scheduling in its general form and the separation that is achieved in the dynamic programming level using Lagrange multipliers. We then apply this general model to obtain a simultaneous beam and waveform scheduling method for radars based on an objective function that depends on both state and action. This method extends on a previous hierarchial method for beam scheduling with an objective function defined only on state. We further improve the objective function based on entropy reduction. This criterion makes the resource management to be more flexible in favor of measurements that carry more information.*

Keywords: Markov decision scheduling, hierarchial control, radar resource management, dynamic programming.

1 Introduction

We are interested fundamentally in the problem of tracking multiple manoeuvring targets with a radar system that can apply different illumination, different waveforms, at each pulse. In this context there are many papers (see e.g. [1, 2, 3, 4]) have produced greedy (one-step ahead) optimal scheduling schemes for both waveforms and beam directions based on various measures of effectiveness of the tracking process across all of the targets. Relatively limited attention has been given to multi-step optimization. This paper undertakes a theoretical analysis of such a scenario by modeling it as a Markov Decision Process (MDP). The individual targets are modeled as independent Markov processes. The measurement of each of those targets is an action of the MDP. The allocation of the measurement resources (and kinds of illumination) across targets with limited available resources is the prime focus of this work. We cast this multi-step scheduling into an adaptive control of multiple Markov chains with a common constraint.

In this paper, we first analyze adaptive control of constrained Multi-object Markov Chains. Single Markov chains with constraint was introduced in [5, 6, 7] and further analyzed in [8]. Here we extend the model to Multi-object Markov Chains where M independent chains evolve based on their corresponding input action, but the control of these chains involves a shared limited resource, hence a common constraint. The Lagrange method can be used to decouple the dynamic programming (DP) optimization method of M chains. This decoupling of DP has been shown previously [9] for classification of M "static" objects where N repeated noisy observations of objects using different sensor modes are scheduled for the minimization of classification error. This renders a two-level hierarchial optimization method where at the dynamic programming level the algorithm determines the optimal actions for each object given the values of Lagrange multipliers. At the higher level, given the DP solutions, it finds the values of Lagrange multipliers which enforce the constraint to be met. Here we use the same approach to a make a hierarchial scheduling method for constrained multi-object Markov chains, where the state of objects evolve over time.

Radar resource management has been studied extensively with various approaches, see e.g. [10] for a survey. Progress in approximate dynamic programming techniques has reduced the computational load of multi-step ahead scheduling and permitted it to be seriously considered for radar resource management. In [11] a hierarchial method has been used in conjunction with DP to solve resource scheduling in a slow-time scale. The multi-step scheduling is similar to solving the constrained multi-object Markov chain where the state of system for every object (which is either a tracking or a searching task) is defined based on the accuracy and continuity of tracks and searches. The action for tracking targets is selection of targets, hence only beam

scheduling is considered. The reward attained in every time epoch is a quality of service measure defined only as a function of state. On the other hand, Information theoretic approaches have been used within the framework of dynamic programming for sensor scheduling and resource management, e.g in [12]. An approach based on an entropy measure, but without DP has been used in [13], where the beam direction (action) is scheduled as optimization of a reward function based on the expected reduction in entropy for various targets. Hence the reward is a function of action. However, the approach is myopic without consideration of future rewards.

In this paper we use the general model of constrained multi-object Markov chains for simultaneous beam and waveform scheduling, where the reward function is considered as a function of both state (accuracy-continuity criteria) and action (due to different reductions in entropy for each action). In our approach both entropy and accuracy-continuity measures are used for scheduling. Hence besides considering the quality of service measure used in [11], the objective function also considers efficiency of measurements and favors updating targets and using waveforms for which more reduction in uncertainty as a result of measurement is expected.

2 Constrained Multi-object Markov Decision Scheduling

2.1 Problem formulation

We consider a Markov decision scheduling for M objects aiming for optimization of the integral of an objective function over a finite horizon N. For clarity, here we indicate the object index by a subscript and the time index by a superscript. The state of each object i, denoted by x_i can assume S possible values and evolves according to a transition probability $P_i(d_i)$ which depends only on the action d_i applied to object i, hence assuming the independence of objects behaviors. Denote the aggregated states and actions as $x = [x_1, \cdots, x_M]$, and $d = [d_1, \cdots, d_M]$, respectively. We define a standard Markov decision process (MDP) with a transition probability matrix $P(d)$, having dimension $S^M \times S^M$ and elements

$$P_{x,x'}(d) = \prod_{i}^{M} P_{i,x_i,x_i'}(d_i).$$

As in the case of MDP's, the aim is to find an optimal policy μ^* that maximizes the expectation of a reward (utility) function $U(x,d) \triangleq \sum_{i=1}^{M} U_i(x_i, d_i)$ over a horizon $k = 1, \cdots, N$, where $U_i(\cdot,\cdot)$ is the object reward function dependent on the state and the action of each of the separate processes. A policy defines what action d^k should be taken given the state x^k at time k. The finite horizon reward is denoted as

$$J(x^0) \triangleq E\{\sum_{k=0}^{N-1} U(x^k, d^k)|x^0\}.$$

However, in contrast to the standard Markov decision scheduling problem, object i consumes a resource $l_{i,k}(x_i^k, d_i^k)$ at time k which in general depends on the state of the object and the action on that object at the time. The total resources consumed by all objects at any given time is limited. Without loss of generality, we assume that the maximum available resource at any time is (normalized to) one, i.e:

$$l_k(x^k, d^k) \triangleq \sum_i l_{i,k}(x_i^k, d_i^k) \leq 1, \forall k.$$

Therefore the Markov decision scheduling requirs optimization of $J(x^0)$ over μ with the above constraint. However, in order to achieve a separation in the problem as explained in Section 2.2, we need to relax the constraints to

$$\bar{l}_k(x^0) \leq 1, \qquad \forall k, \tag{1}$$

where

$$\bar{l}_k(x^0) = E^*_{x^k|x^0} l_k(x^k, d^{*k}(x^k)). \tag{2}$$

Here $d^{*k}(x^k)$ is the optimal action for x^k based on the optimal policy under the constraint, and the notation $E^*_{x^k|x^0}$ means the expectation with respect to x^k conditioned on x^0 assuming the optimal policy is used up to time $k-1$, i.e.:

$$E^*_{x^k|x^0} f(x^k) \triangleq E_{x_1|x^0, d^{*0}(x^0)}\{... \\ E_{x^{k-1}|x^{k-2}, d^{*k-2}(x^{k-2})}\{E_{x^k|x^{k-1}, d^{*k-1}(x^{k-1})} f(x^k)\}...\} \tag{3}$$

Note $\bar{l}_0(x^0) = l_0(x_0, d^{*0}(x^0))$, hence for $k = 0$, the constraint (1) is deterministic, but for $k > 0$ it is a constraint on the average over all possible sample paths in an optimal closed loop control.

Without the constraint, because of the local action of the d_i on the x_i the problem was that of scheduling M separate Markov decision processes. But with the constraint the problem requires allocation of a fixed resource across each of the separate processes., i.e: action d_i needs to be chosen with the consideration of other actions. Hence, the problem is large-scale over the vector values x and d. We note that the dimension of state Markov process (hence the size of Matrix $P(d)$) grows exponentially with number of objects M, hence the large-scale problem is computationally infeasible to solve for large S and M. In the next section we show an approach for breaking the problem into a set of small-scale problems.

2.2 Solution by decomposition using Lagrange multipliers

Instead of the above MDP for finding an optimal policy we solve the following optimization problem over d^0

$$\max_{d^0} J(x^0) \tag{4}$$

with the set of constraints (1). This is a simpler version of the problem which only finds the optimal action in the start for a given initial state x^0, but by solving it for every x^0 we obtain an optimal policy μ_0^*, or in turn a stationary policy μ^*. If there was no constraint, then the problem (4) could be written as

$$\max_{d^0}\{U(x^0,d^0)+E_{x^1|x^0,d^0}J_1^*(x^1)\}, \tag{5}$$

where $J_1^*(x^1)$ represents the optimal value-to go (optimal expected reward that can be obtained starting from x^1). This in turn relates to a sequence of optimal future value-to-go functions, i.e:

$$J_k^*(x^k)=\max_{d^k}\{U(x^k,d^k)+E_{x^{k+1}|x^k,d^k}J_{k+1}^*(x^{k+1})\}. \tag{6}$$

In fact the optimal objective function in (4) with no constraint is equal to $J_0^*(x^0)$. Note that in the nested optimizations $J_k^*(x^k)$, x^k is not an optimization variable, but it can be considered as a given parameter.

Using Lagrange relaxation, we include the constraints in the optimization (4) as

$$\max_{d^0,\lambda^0,\ldots,\lambda^{N-1}} J(x^0)+\sum_{k=0}^{N-1}\lambda^k(1-\bar{l}_k(x^0))$$

This optimization can be written as

$$\max_{d^0,\lambda^0}\{U(x^0,d^0)+\lambda^0(1-l_0(x^0,d^0)) + E_{x^1|x^0,d^0}\max_{d^1,\lambda^1}L_1(x^1,d^1,\lambda^1)\}, \tag{7}$$

where

$$L_1(x^1,d^1,\lambda^1)=U(x^1,d^1)+\lambda^1(1-l_1(x^1,d^1)) + E_{x^2|x^1,d^1}\max_{d^2,\lambda^2}L_2(x^2,d^2,\lambda^2), \tag{8}$$

and sequentially,

$$L_k(x^k,d^k,\lambda^k)=U(x^k,d^k)+\lambda^k(1-l_k(x^k,d^k)) + E_{x^{k+1}|x^k,d^k}\{\max_{d^{k+1},\lambda^{k+1}}L_{k+1}(x^{k+1},d^{k+1},\lambda^{k+1})\}. \tag{9}$$

Therefore, $L_k(x^k,d^k,\lambda^k)$ at the optimal d^k,λ^k represents the optimal value-to-go for any given x^k taking into account the resource constraint. We denote this optimal value-to-go with constraint by $\hat{J}_k^*(x^k)$. At the last stage N, $L_N(x^N)=U(x^N)=\sum_{i=1}^M U_i(x_i^N)$; there is no action at time N. Note that the optimal λ^k and d^k depends on x^k as a given parameter. Denoting the optimal λ^k and d^k by $\lambda^{*k}(x^k), d^{*k}(x^k)$, respectively, we can write $L_k(x^k,d^k,\lambda^k)$ as

$$L_k(x^k,d^k,\lambda^k)=U(x^k,d^k)+\lambda^k(1-l_k(x^k,d^k)) + E_{x^{k+1}|x^k,d^k}\{\max_{d^{k+1}}L_{k+1}(x^{k+1},d^{k+1},\lambda^{*k+1}(x^{k+1}))\}. \tag{10}$$

Here λ^k is the Lagrange multiplier at time k (as one of the optimization variables in the nested optimizations), but $\lambda^{*k}(x^k)$ is the Lagrange multiplier such that the resource constraint is fulfilled with equality at optimum $d^{*k}(x^k)$. Also, as was remarked earlier

$$\hat{J}_k^*(x^k)=L_k(x^k,d^{*k}(x^k),\lambda^{*k}(x^k)).$$

The trick for decomposition of the problem into separate ones here is to replace $\lambda^{*k}(x^k)$ in (10) for various k by an estimate $\hat{\lambda}=[\hat{\lambda}^0,\cdots,\hat{\lambda}^{N-1}]$ which we assume that we have obtained such that (1) is satisfied with equality, i.e.:

$$E^*_{x^k|x^0}\{l_k(x^k,d^{*k}(x^k))\}=1. \tag{11}$$

Proposition 1 *Given a fixed vector $\hat{\lambda}$, the lagrangian $L_k(x^k,d^k,\lambda^k)$ is separable as*

$$L_k(x^k,d^k,\hat{\lambda})=\sum_{i=1}^M L_{i,k}(x_i^k,d_i^k,\hat{\lambda})+\sum_{n=k}^{N-1}\hat{\lambda}^n \tag{12}$$

where

$$L_{i,k}(x_i^k,d_i^k,\hat{\lambda})=U_i(x_i^k,d_i^k)-\hat{\lambda}^k l_{i,k}(x_i^k,d_i^k) + E_{x^{k+1}|x^k,d^k}\{\max_{d^{k+1}}L_{i,k+1}(x_i^{k+1},d_i^{k+1},\hat{\lambda})\}. \tag{13}$$

Proof 1 *Equation* (12) *is true for $k=N$ by assigning $L_{i,N}(x^N)=U_i(x^N)$. Assuming*

$$L_{k+1}(x^{k+1},d^{k+1},\hat{\lambda})=\sum_{i=1}^M L_{i,k+1}(x_i^{k+1},d_i^{k+1},\hat{\lambda})+\sum_{n=k+1}^{N-1}\hat{\lambda}^n \tag{14}$$

we have from (10),

$$L_k(x^k,d^k,\hat{\lambda})=\sum_{i=1}^M(U_i(x_i^k,d_i^k)-\hat{\lambda}^k l_{i,k}(x_i^k,d_i^k))+\hat{\lambda}^k + E_{x^{k+1}|x^k,d^k}\{\max_{d^{k+1}}\sum_{i=1}^M L_{i,k+1}(x_i^{k+1},d_i^{k+1},\hat{\lambda})\}+\sum_{n=k+1}^{N-1}\hat{\lambda}^n. \tag{15}$$

But because of the local influence of d_i only on L_i we can interchange the sum and max operations

$$L_k(x^k,d^k,\hat{\lambda})=\sum_{i=1}^M(U_i(x_i^k,d_i^k)-\hat{\lambda}^k l_{i,k}(x_i^k,d_i^k)) + E_{x^{k+1}|x^k,d^k}\{\sum_{i=1}^M\max_{d^{k+1}}L_{i,k+1}(x_i^{k+1},d_i^{k+1},\hat{\lambda})\}+\sum_{n=k}^{N-1}\hat{\lambda}^n. \tag{16}$$

This reads again as

$$L_k(x^k, d^k, \hat{\lambda}) = \sum_{i=1}^{M}(U_i(x_i^k, d_i^k) - \hat{\lambda}^k l_{i,k}(x_i^k, d_i^k) + E_{x^{k+1}|x^k,d^k}\{\max_{d^{k+1}} L_{i,k+1}(x_i^{k+1}, d_i^{k+1}, \hat{\lambda})\}) + \sum_{n=k}^{N-1} \hat{\lambda}^n. \quad (17)$$

which proves (12) *for any* k.

Now Proposition 1 shows that solving the value function $\hat{J}^*(x^k)$ is separable into M optimizations,

$$\begin{aligned}\hat{J}^*(x^k) &= \max_{d^k} L_k(x^k, d^k, \hat{\lambda}^k) \\ &= \sum_{n=k}^{N-1} \hat{\lambda}^n + \max_{d^k} \sum_{i=1}^{M} L_{i,k}(x_i^k, d_i^k, \hat{\lambda}) \\ &= \sum_{n=k}^{N-1} \hat{\lambda}^n + \sum_{i=1}^{M} \max_{d_i^k} L_{i,k}(x_i^k, d_i^k, \hat{\lambda}) \quad (18)\end{aligned}$$

The third equality is because $L_{i,k}$ only contains d_i component of d. This significantly simplifies the computation of optimal $d^0 = [d_1^0, \cdots, d_M^0]$ for a given x^0, i.e: solving for $\hat{J}^*(x^0)$ which is the objective of (4) with constraints. However the price is that we need extra recursive computation for updating estimates of $\hat{\lambda}$. Using a first order approximation for (23), we can update the estimate of $\hat{\lambda}^k$ to $\hat{\lambda}_u^k$ by

$$\hat{\lambda}_u^k = \hat{\lambda}^k + \Delta\hat{\lambda}^k \quad (19)$$

where $\Delta\hat{\lambda}^k$ is chosen such that

$$\bar{l}_k(x^0) + \frac{\partial \bar{l}_k(x^0)}{\partial \hat{\lambda}} \Delta\hat{\lambda} = 1,$$

for every k. Here $\bar{l}_k$ (and $\partial\bar{l}_k/\partial\hat{\lambda}$) need to be calculated (estimated) from (2) using the optimal d^{*k} obtained from the solution of $\hat{J}^*(x^k)$. We then solve for $\hat{J}^*(x^k)$ using the new estimate $\hat{\lambda}_u^k$, and repeat the two operations until satisfactory convergence is observed.

With the above alternative update of $\hat{\lambda}$ and d^*, the original problem is decoupled hierarchically into two levels as explained before. In the dynamic programming level a separation into the large-scale control problem is achieved, where given the value of Lagrange multipliers, each component is optimized locally as a small scale problem. However, the components are coordinated globally via the constraint and the component competition for the resources. Here we discuss a simple application of the constraint MDP model for which this solution approach can be used. This example helps in understanding the radar resource management problem in Section 3.

2.3 Example: Application to multi-project time management

As an example we apply the above model to find the optimal percentage of time one should spend on each project from a list of M projects in a period of time, say every week, given the status of projects at the start of the period. Suppose the performance status for project i is indicated by a number $x_i \in \{1, \cdots, S\}$. Status 1 means the project is running very well while S indicates the worst performance progress. Based on priorities and importance of projects, one assigns a reward function $U_i(x_i)$ to each project, and the instantaneous reward (achieved in one week) is $U(x) = \sum_i U_i(x_i)$, where $x = [x_1, \cdots, x_i]$. The aim of scheduling is to find what percentage of time d_i should be spend on each project i, aggregated in $d = [d_1, d_2, \cdots, d_M]$, such that the reward over a horizon of N weeks be maximized, i.e. find the optimal d^0 that solves

$$\max_{d^0} E\{\sum_{n=0}^{N-1} U(x^n)|x^0\}$$

where x^k (x at week k) evolves as a Markov process with a transition probabilities depending on what time percentage d^k is spent at week k. The above optimization problem is constrained to

$$l_k(d^k) \triangleq \sum_i d_i^k \leq 1, \forall k \quad (20)$$

This constraint is because the total percentage of time spent over all projects in a week cannot be greater than one.

With the above description, the project management is a constrained multi-object Markov decision scheduling, where the resource consumed by every object is simply the percentage of time allocated to that project, i.e.:

$$l_{i,k}(x_i^k, d_i^k) = d_i^k$$

For the above problem the dynamics governing the change of status of the projects in one week based on the time spent on the project needs to be defined through a transition probability matrix. Assuming the project dynamics are independent, the transition probabilities can be defined for each project i separately. For example, the transition probability $P_i(x_i^{k+1}|x_i^k)$ can be defined parametricaly as

$$P_i(x_i^{k+1}|x_i^k) = \begin{cases} a_i(d_i^k) & x_i^{k+1} = x_i^k; \\ b_i(d_i^k) & x_i^{k+1} = x_i^k + 1; \\ c_i(d_i^k) & x_i^{k+1} = x_i^k - 1. \end{cases}$$

where $a_i(d_i^k) + b_i(d_i^k) + c_i(d_i^k) = 1$ when $1 < x_i < S$, and for $x_i = 1$ and $x_i = S$, in addition $c_i(d_i^k) = 0$ and $b_i(d_i^k) = 0$, respectively.

To solve this problem based on the method in Section 2.2, we need to relax the constraint (20) to (1). So

given the status of each project at the start of week, aggregated in x^0, one can solve problem (4) to find the best percentage of time that should be spent on each project at that week, i.e. optimal d^0.

3 Application to Radar resource management

In similar vein to the above project management problem, here we use the constrained multi-object Markov chains for radar resource scheduling. A similar approach has been used in [11] for this scheduling by considering two time-scales. The scheduling needs to be done periodically with a slow time scale period of Δ, say $\Delta = 1$sec. The fast time scale is used for managing the operation of tasks scheduled in the slow time-scale. In the slow-time scale scheduling, instead of projects in the above example, there are large number of tasks that need to be prioritized and scheduled. Tasks are of tracking or searching type. A tracking task is an operation that update a target position and a searching task is an operation that search a sector of space for new targets. Two different state variables for each target i is defined. The kinematic state ξ_i is the actual position and velocity of target. But of primary importance is x_i, the state of the task dealing with target i, including the accuracy and continuity of tracking the target.

Here we assume that the scheduling outcome for each period is a set of numbers $d_i^k \in \{0, 1, \cdots, D\}$, where $d_i^k \neq 0$ means that the task i to be performed with measurement (waveform or illumination) index d_i^k, otherwise $d_i^k = 0$. The binary case $D = 1$ corresponds to beam only scheduling (the problem dealt with in [11]). The resource that is consumed by each task is time. We assume that each task i takes a deterministic time $T_i(x_i)$. The normalized time (with respect to Δ) that is consumed on task i is therefore

$$l_{i,k}(x_i^k, d_i^k) = \frac{1}{\Delta} B(d_i^k) T_i(x_i^k),$$

where $B(d)$ is one for $d > 0$, otherwise it is zero. Given the set of decisions $d^k = [d_1^k, \cdots, d_M^k]$ in a period k about which tasks to be performed, the total fraction of slow time scale taken to perform all tasks is $l_k(x^k, d^k) = \sum_i l_{i,k}(x_i^k, d_i^k)$ which needs to be no greater than one. The radar resource management is therefore a Multi-object Markov Decision Scheduling with the set of constraints $l_k(x^k, d^k) \leq 1, \forall k$. However, in order to use the method in Section 2.2, we need to relax the constraint to (1)[1].

[1]In contrast to our setup, in [11] the time required by task, $T_i(x_i)$ is a random variable, and therefore the normalized time that may be taken by task i is on average $\bar{l}_{i,k}(x_i^k, d_i^k) = 1/(\Delta) d_i^k \mathbb{E}\{T_i(x_i^k)\}$, for $D = 1$. The constraint that is enforced for the scheduling is based on this $\bar{l}_{i,k}(x_i^k, d_i^k)$ (according to [11, Eq. 13], and the note that the expectation is not over state), yet $\bar{l}_{i,k}(x_i^k, d_i^k)$ appears in their separable Lagrangian. Nevertheless, an expression similar to $\bar{l}(x_0)$ in (2) is used for the update of $\hat{\lambda}$ in (19) (see [11, Eq.s 20, 25]).

The tracking type tasks are Markov processes where the state of the task x_i mainly defines the accuracy of the tracking. The track accuracy is parameterized by the time intervals (digitized by number of Δ times slots) between the last two track updates.

As is done in [11], we consider the following state vector for target i at time k,

$$x_i^k = (k - k_m, k_m - k_{m-1}, x_{tracked}^k, x_{droped}^k, x_{reint}^k, x_{mix}^k)_i$$

where $(k_m)_i$ means the time index k that the update number m is fulfilled for target i, assuming the last update number is m. All the last four components of x_i^k are binary, where the first three binary components identify the track continuity, and the last one identifies correct association of track with target i. We use a bracket notation [] to denote the components of x.

The state transition probability matrix for every target is a sparse matrix. The non-zero components are defined by the following probabilities,

$$\begin{aligned} &Pr\{x_i^{k+1} | x_i^k[1] \neq K, x_i^k[3] = 1, d_i^k \neq 0\} \\ &= \begin{cases} 1 - p_i(d_i^k), & (x_i^k[1] + 1, x_i^k[2], 1, 0, 0, 0) \\ p_i(d_i^k) p_{c_i}, & (1, x_i^k[1], 1, 0, 0, 0) \\ p_i(d_i^k)(1 - p_{c_i}), & (1, x_i^k[1], 1, 0, 0, 1) \end{cases} \end{aligned} \tag{21}$$

where $p_i(d)$ is the probability of detecting the i-th target signal in the operation of updating the target position using the waveform index d, and p_{ci} is the probability of correct association for target i. If the target is detected, $k - k_m$ becomes $k_m - k_{m-1}$ for the state in the next epoch. But if in spite of the command to update the target position, $d_i^k \neq 0$ the target is not detected, then the time from last update, i.e.: $x_i[1]$ increases by one. Here, K is the maximum time elapse that we consider with no detection before the track drops. For $x_i^k[1] = K$,

$$\begin{aligned} &Pr\{x_i^{k+1} | x_i^k[1] = K, x_i^k[3] = 1, d_i^k \neq 0\} \\ &\qquad = \begin{cases} 1 - p_i(d_i^k), & (0, 0, 0, 1, 0, 0) \\ p_i(d_i^k), & (1, K, 1, 0, 0, 0) \end{cases} \end{aligned}$$

For $d_i^k = 0$ the transition probabilities are the same as above when replacing $p_i(d_i^k) = 0$. We also have

$$Pr\{x_i^{k+1} | x_i^k[3] = 0\} = \begin{cases} 1 - p_{d,search}, & (0, 0, 0, 0, 0, 0) \\ p_{d,search}, & (0, 0, 1, 0, 0, 0) \end{cases}$$

$$Pr\{x_i^{k+1} | x_i^k[4] = 1\} = \begin{cases} 1 - p_{d,search}, & (0, 0, 0, 1, 0, 0) \\ p_{d,search}, & (0, 0, 1, 0, 1, 0) \end{cases}$$

where $p_{d,search}$ is the search scan detection probability given waveform index d, otherwise it is zero. For

the relation of $p_i(d_i^k), p_{ci}, p_{d,search}$ with the system and target characteristics refer to [11].

Next we define the reward function for the simultaneous beam and waveform scheduling. This is a function of both state and action. The utility function defined in [11] is a special cased of this reward function for the beam only scheduling, where the reward function reduces to a function of state only. We then improve the reward function to include the information efficiency of measurements based on entropy function. Having defined the reward functions $U_i(x_i^k, d_i^k), i = 1, \cdots, M$ as a combined criteria, the objective is to solve

$$\max_{d^0} E\{\sum_{n=0}^{N-1} U(x^n, d^n)|x^0\},$$

under the constraint (1). At the start of each period, given the state of tasks x^0 the optimal solution d^0 identifies which targets need to be looked at in that period using what waveform indices.

3.1 Reward function for simultaneous beam and waveform scheduling

Assume the following linear Gaussian model for the state ξ_i and observation y_i of the ith target holds.

$$\xi_i(t+T) = F(T)\xi_i(t) + w_i(T)$$

$$y_i(t) = H_i(d_i^t)\xi_i(t) + v_i(d_i^t), \tag{22}$$

where w_i and v_i are the zero mean Gaussian state and measurement noises with covariance matrices $Q_i(T), R_i(d_i^t)$, respectively. Note here the measurement from the target (both the linear map component and the noise covariance) depends on the waveform index d_i selected at time t.

Given the error covariance matrix at update time k_{m-1}, denoted by $P_i^{k_{m-1}|k_{m-1}}$, we can obtain $P_i^{k|k_m}$ at update time k using two consecutive steps of the Kalman filter. This will be based on the time intervals $\delta_{i,1} = (k - k_m)\Delta, \delta_{i,2} = (k_m - k_{m-1})\Delta$ and action d_i^k for target i, using the approximation that two consecutive applied waveforms characteristics do not change significantly and they have almost the same effect on measurement.

$$\begin{aligned} P_i^{k_m|k_{m-1}} &= F(\delta_{i,2})P_i^{k_{m-1}|k_{m-1}}F(\delta_{i,2})^T + Q_i(\delta_{i,2}) \\ P_i^{k_m|k_m} &\simeq (I - K_{i,m-1}H_i(d_i^k))P_i^{k_m|k_{m-1}} \\ &\quad \times (I - K_{i,m-1}H_i(d_i^k))^T \\ &\quad + K_{i,m-1}R_i(d_i^k)K_{i,m-1}^T \\ P_i^{k|k_m} &= F(\delta_{i,1})P_i^{k_m|k_m}F(\delta_{i,1})^T + Q_i(\delta_{i,1}) \end{aligned} \tag{23}$$

Here $K_{i,n}$ is the Kalman gain at update time n. Therefore knowing $P_i^{k_{m-1}|k_{m-1}}$ from past observations, we can obtain $P_i^{k|k_m}$ matrix as a deterministic function of $x_i[1] = \delta_{i,1}/\Delta$, $x_i[2] = \delta_{i,2}/\Delta$ and d_i^k. The scalar error variance is then defined as

$$\sigma_i^k = \sqrt{H_i(d_i^k)P_i^{k|k_m}H_i(d_i^k)^T}$$

which is a deterministic function of $x_i^k[1], x_i^k[2], d_i^k$. For a given state x_i calculation of σ_i^k requires a parameter $P_i^{k_{m-1}|k_{m-1}}$, which is considered based on an average update rate for target i in the recent past.

We define the reward function as an extension of the one used in [11] by

$$U_i(x_i^k, d_i^k) = \alpha Q_{acc}(\sigma_i)x_i^k[3] - C_{reinit}x_i^k[5] - C_{mix}x_i^k[6], \tag{24}$$

where α is a nominal utility measure for tracking which may also correspond to user priority, Q_{acc} is a decreasing function with a range $[0, 1]$, and C_{reinit} and C_{mix} are the cost for a track reinitiation and mix. In contrast to the utility function in [11], here σ_i requires both state and the waveform index d_i^k.

3.2 Improving the reward function in favor of uncertainty reduction

The covariance $P_i^{k|k_m}$ in (23) is the predicted covariance for time k given the history of measurements from target i without any measurement at time k. If we get a measurement at time k from this target, i.e. choose $d_i^k \neq 0$ and be able to detect the target, then the measurement will change the covariance of state estimate to $P_i^{k|k}$ that can be obtained from the second equation in (23). The determinant of $P_i^{k|k_m}$ represent the uncertainty about the target at time k without measurement and the determinant of $P_i^{k|k}$ represents expected uncertainty with the measurement at time k which depends on the applied waveform index d_i^k. The accuracy of system as a whole considering all targets will be further improved in the radar resource management if we favor choosing the targets for update that makes more reduction in the uncertainty as a result of next measurement. Here we modify the utility function (24) to include this uncertainty reduction. The utility function (24) has only parameterized $P_i^{k|k_m}$, the uncertainty that we are in at time k, but not how efficient a measurements from target i will be at this time.

Here we assume that the kinematic state of the targets has azimuthal and elevation coordinates that for i-th target we denote by ϕ_i, r_i, respectively, Also let the predicted variance for azimuth and elevation estimation at time k be $\sigma_{i\phi}(k|k_m), \sigma_{i,r}(k|k_m)$. For the special case when the target azimuth and elevation coordinates are independent, the Kalman filter equations decouple, and each coordinate can be updated and predicted separately. In this case, $|P_i^{k|k_m}| = \sigma_{i\phi}(k|k_m)\sigma_{i,r}(k|k_m)$.

The predicted variances $\sigma_{i\phi}(k|k_m), \sigma_{i,r}(k|k_m)$ can be obtained from the updated variances

$\sigma_{i,\phi}(k_m|k_m), \sigma_{i,\phi}(k_m|k_m)$ based on the target dynamics using the following equations [13],

$$\sigma^2_{i,\phi}(k|k_m) = \sigma^2_{i,\phi}(k_m|k_m) + 2\zeta_{i,\phi}\delta_{i,1} + \eta_{i,\phi}\delta^2_{i,1}$$

$$\sigma^2_{i,r}(k|k_m) = \sigma^2_{i,r}(k_m|k_m) + 2\zeta_{i,r}\delta_{i,1} + \eta_{i,r}\delta^2_{i,1}$$

where $\zeta_{i,\pi}$, $\zeta_{i,r}$ are the position/velocity covariances for the ϕ_i and r_i variables and $\eta_{i,\phi}, \eta_{i,r}$ are the velocity variances. ($\delta_{i,1}$ was defined in (23)). For a given state x_i (note $\delta_{i,1} = \Delta x_i[1]$), $\sigma^2_{i,\phi}(k|k_m)$ can be calculated based on the parameter $\sigma^2_{i,\phi}(k_m|k_m)$ which (similar to calculation of $P_i^{k|k_m}$) is considered based on an average update rate for target i.

We consider entropy as the measure of uncertainty. Because of the Gaussian assumption on the dynamical process noise, the entropy of the kinematic state of the i-th target at time k without measurement is, [14]

$$\begin{aligned} h_i(k) &= 1/2\log(4\pi^2 e^2 |P_i^{k|k_m}|) \\ &= \log 2\pi e + 1/2(\log\sigma_{i\phi}(k|k_m) + \log\sigma_{ir}(k|k_m)) \end{aligned} \quad (25)$$

Here, we use the symbol σ_i as a generic symbol for both $\sigma_{i,\phi}, \sigma_{i,r}$ to avoid repeating similar equations resulting from the decoupled Kalman filter. If we choose the action $d_i^k \neq 0$ to get an update for i-th target, and this target is detected, then the Kalman filter with measurement update gives a new estimate with covariance $\sigma_i(k|k)$ satisfying [13]

$$\frac{1}{\sigma_i^2(k|k)} = \frac{1}{\sigma_i^2(k|k_m)} + \frac{1}{\bar{\sigma}_i^2(d_i^k)}$$

where $\bar{\sigma}_i^2(\cdot)$ is the variance of the measurement noise, obtained from $R_i(\cdot)$, and both are functions of the waveform used for the measurement. The probability of detecting the target however is equal to Pr{target detected $\|$ target in the beam}Pr{target in the beam}=$P_D P_B$ where $P_D = (P_{FA})^{1/(1-SNR)}$ for a swerling type I fluctuating target and P_B can be estimated by integrating the target probability distribution over the 3dB pencil beam [13]. With probability $(1 - P_D P_B)$ the target is not detected and with no observation, there will be no update, hence, $\sigma_i(k|k) = \sigma_i(k|k_m)$. As a result, the expected entropy assuming $d_i^k \neq 0$ is

$$\bar{h}_i(k) = \log 2\pi e + 1/2E(\log\sigma_{i\phi}(k|k) + \log\sigma_{ir}(k|k)).$$

Therefore,

$$\begin{aligned} \bar{h}_i(k) &= \log 2\pi e + 1/4 P_D P_B \times \\ &(-\log(\frac{1}{\sigma^2_{i,\phi}(k|k_m)} + \frac{1}{\bar{\sigma}^2_{i,\phi}(d_i^k)}) - \log(\frac{1}{\sigma^2_{i,r}(k|k_m)} + \frac{1}{\bar{\sigma}^2_{i,r}(d_i^k)})) \\ &+ 1/2(1 - P_D P_B)(\log\sigma_{i\phi}(k|k_m) + \log\sigma_{ir}(k|k_m)). \end{aligned} \quad (26)$$

Comparing (26) with the entropy in the absence of measurement (25), we see that the expected reduction in entropy as a result of action $d_i^k \neq 0$ is

$$\begin{aligned} H_{\Delta,i} &= -1/4 P_D P_B \\ &\times(\log\frac{\bar{\sigma}^2_{i,\phi}(d_i^k)}{\sigma^2_{i,\phi}(k|k_m) + \bar{\sigma}^2_{i,\phi}(d_i^k)} + \log\frac{\bar{\sigma}^2_{i,r}(d_i^k)}{\sigma^2_{i,r}(k|k_m) + \bar{\sigma}^2_{i,r}(d_i^k)}). \end{aligned} \quad (27)$$

This reduction in entropy (positive quantity) can be considered as a reward for action $d_i^k \neq 0$.

By considering the reward $H_{\Delta,i}$ in the utility function, the scheduling also becomes in favor of selecting the targets that have the largest predicted covariances ($\sigma_i^2(k|k_m)$)and the largest probabilities of being detected ($P_D P_B$), and at the same time trying to reduce the measurement error ($\bar{\sigma}_i^2(d_i^k)$) by selecting a suitable waveform for that target state. This very much conforms with a natural target and waveform selection criteria. In effect, the factor $H_{\Delta,i}$ encourages the actions that reduces the uncertainty in the target estimation. Consequently, the following utility function for radar resource scheduling encapsulates all criteria for track accuracy, continuity, and resource efficiency,

$$\begin{aligned} U_i(x_i^k, d_i^k) &= \alpha Q_{acc}(\sigma_i) x_i^k[3] + \beta H_{\Delta,i} B(d_i^k) \\ &- C_{reinit} x_i^k[5] - C_{mix} x_i^k[6], \end{aligned} \quad (28)$$

where β is a nominal factor for prioritizing the uncertainty factor, also depends on the unit of entropy. Note $H_{\Delta,i}$ is a function of both state x_i and action d_i.

4 Conclusion

In this paper we established the mathematical model for the constrained multi-object Markov decision scheduling as an extension of single ones. The large scale dynamic programming required for the multiple Markov chains competing for a limited resource turns to multiple small-scale optimization using Lagrange method. This mathematical model and its solution methodology can be used for any multi-task (multi-project) scheduling considering time as the available resource. A variation of this model has already been used in radar resource management for beam scheduling where the objective function is only a function of state of tasks. Here we used this model for simultaneous beam and waveform scheduling. In this application in accordance with the general model of constrained multi-object Markov chains the objective function is a function of both state and action. We further improved the scheduling criterion based on an entropy measure. The use of entropy criterion introduced a weighted term in the objective function in favor of measurements that are more likely to reduce uncertainty in the whole set of targets. The added term encourages actions (targets to be selected and the the waveform to be used) based on criterion that conforms well with intuition, i.e.: targets

to be selected that have the largest estimation covariances and most likely to be detected, and the waveforms that have the smallest measurement error under current conditions.

Acknowledgement

This research was supported by Australian Research Council Discovery grant number DP0878269.

References

[1] D. Cochran, S. Suvorova, S. Howard, and B. Moran, "Waveform Libraries", IEEE Signal Processing Magazine, vol. 26, No. 1, 2009.

[2] B. La Scala, M. Rezaeian, and B. Moran, "Optimal Adaptive Waveform Scheduling for Target Tracking", in proceedings of International Conference on Information Fusion, Philadelphia, PA, July 2005.

[3] S. Suvorova and D. Musicki and B. Moran and S. Howard and B. La Scala, "Multi step ahead beam and waveform scheduling for tracking of manoeuvering targets in clutter", Int. Conf. Acoust., Speech, Signal Processing, Philadelphia, PA, March 2005.

[4] B. Moran and S. Suvorova and S. Howard, "Application of Sensor Scheduling Concepts to Radar", Book chapter, A.O Hero III, D.A. Castanon, D. Cochran, and K. Kastella, "Foundations and Applications of Sensor Management", Springer, 2007.

[5] F. Beutler, K. Ross, "Optimal Policies for Controlled Markov Chains with a Constraint", Journal of Mathematical Analysis and Applications, vol. 112, pp 236-252, 1985.

[6] D. Ma, A.Makowski and A. Shwartz, "Estimation and optimal control for constrained Markov chains", Proceedings of 25th conference on Decision and Control, Dec. 1986.

[7] E. Altman and A. Shwart, "Adaptive Control of Constrained Markov Chains", IEEE Transactions on Automatic Control, Vol. 36, NO. 4, April 1991.

[8] A. Zadorojniy and A. Shwartz, "Robustness of Policies in Constrained Markov Decision Processes", IEEE Transactions on Automatic Control, Vol. 51, NO. 4, April 2006.

[9] D. Castanon, Approximate dynamic programming for sensor management", In Proceedings of the 36th Conference on Decision and Control, 1997.

[10] Z. Ding, "A survey of Radar resource management algorithms", IEEE CCECE/CCGEI, Canada, 2008.

[11] J. Wintenby, V. Krishnamurthy, "Hierarchical Resource Management in Adaptive Airborne Surveillance Radars", IEEE Transactions on Aerospace and Electronic Systems, Vol. 42, NO. 2 April 2006.

[12] M. Rezaeian, "Sensor Scheduling for Optimal Observability Using Estimation Entropy", workshop proceedings of 5th IEEE International Conference on Pervasive Computing and Communications, New York, March 2007.

[13] P. Berry, D. Fogg, "On the Use of Entropy for Optimal Radar Resource Management and Control" IEEE Radar Conference, 2003.

[14] T.M.Cover and J.A.Thomas. "Elements of Information Theory", Wiley, New York, 1991.