Available online at www.sciencedirect.com

ScienceDirect

IFAC-PapersOnLine 49-17 (2016) 007–010

Optimal Threshold Policy for Sequential Weapon Target Assignment

Krishnamoorthy Kalyanam * **Sivakumar Rathinam** ** **David Casbeer** *** **Meir Pachter** ****

* *InfoSciTex Corporation, a DCS Company, Dayton, OH 45431 USA (e-mail: krishnak@ucla.edu).*
** *Mechanical Engineering Department, Texas A&M University, TX 77843 USA (e-mail: srathinam@tamu.edu)*
*** *Autonomous Control Branch, Air Force Research Laboratory, WPAFB, OH 45433 USA (e-mail: david.casbeer@us.af.mil)*
**** *Electrical & Computer Engineering Department, Air Force Institute of Technology, WPAFB, OH 45433 USA (e-mail: meir.pachter@afit.edu)*

Abstract: We investigate a variant of the classical Weapon Target Assignment (WTA) problem, wherein N targets are sequentially visited by a bomber carrying M homogenous weapons. We are interested in the optimal assignment of weapons to targets. A weapon launched by the bomber destroys the j^{th} target with probability p_j and upon successful elimination, the bomber earns a positive reward r_j. There is feedback in that the bomber, upon deploying a weapon, is notified whether or not it successfully destroyed the target. Whereupon, it decides whether to move on to the next target or allocate an additional weapon to the current target. We provide a tractable solution method for computing the optimal closed loop control policy that results in maximal expected total reward. Moreover, we show that a thresholding policy is optimal, wherein a weapon from an inventory of k weapons is dropped on the j^{th} target iff k exceeds a stage dependent threshold value, $\kappa(j)$.

Keywords: Weapon-Target Assignment, Dynamic Programming, Threshold Policy

1. INTRODUCTION

The operational scenario is the following. A bomber with M identical weapons travels along a designated route/ path and sequentially encounters enemy (ground) targets numbered 1 to N. A weapon dropped on the j^{th} target will destroy it with probability p_j. Upon successful elimination, the bomber receives a positive reward r_j. Furthermore, upon releasing a weapon, the bomber is alerted as to whether or not the deployed weapon was successful. If the engagement was successful, the bomber moves on to the next target. If it was not successful, the bomber can either re-engage the current target or move on to the next target in the sequence. We compute the optimal feedback policy that results in maximal total expected reward for the bomber.

2. WEAPON-TARGET ASSIGNMENT PROBLEM

Upon inspecting the scenario considered herein, it is clear that, if there was no feedback, the problem collapses to the static Weapon-Target Assignment (WTA) problem. Indeed, in the absence of feedback information, the bomber might as well decide how many weapons to be dropped on each target, at the start of the mission. Towards that end, let n_j be the number of weapons assigned to target j. The expected reward from this assignment is given by:

$$\bar{V}_j(n_j) = r_j(1 - q_j^{n_j}), \tag{1}$$

where $q_j = 1 - p_j$. So, the assignment problem is given by:

$$\max \sum_{i=1}^{N} \bar{V}_i(n_i) \text{ subject to} \tag{2}$$

$$\sum_{j=1}^{N} n_j = M \text{ and } n_j \in \{0, \ldots, M\}, \forall j. \tag{3}$$

Alternatively, one can rewrite the above as a minimization problem:

$$\min \sum_{i=1}^{N} r_j q_j^{n_j} \text{ subject to} \tag{4}$$

$$\sum_{j=1}^{N} n_j = M \text{ and } n_j \in \{0, \ldots, M\}, \forall j. \tag{5}$$

The above problem is a special case of Flood's static Weapon-Target Assignment (WTA) problem - see Manne (1958) and the optimal solution is obtained via the Maximum Marginal Return (MMR) algorithm - see denBroeder et al. (1959).

Algorithm MMR

1. Initialize $n_j = 0$ and $V_j = r_j \; \forall j = 1, \ldots, N$
2. for $i = 1, \ldots, M$
3. Find $k = \arg\max_{j=1}^{N} p_j V_j$
4. Update $n_k = n_k + 1$ and $V_k = V_k q_k$

Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2016.09.002

Algorithm MMR has a time complexity of $\mathcal{O}(N + M\log(N))$. Note that under the additional complexity that the weapons are not homogenous i.e., if p_{ij} indicates the probability that target j gets destroyed upon being hit by weapon of type i, the resulting static assignment problem is NP-complete - see Lloyd and Witsenhausen (1986). Exact and heuristic algorithms to solve this version of the WTA problem are provided in Ahuja et al. (2007). An approximate algorithm for a dynamic WTA problem, wherein not all targets are known to the decision maker at the start of the mission, is provided in Murphey (1999). A geometry based optimal weapon target assignment for convoy protection is provided in Leboucher et al. (2013).

3. MODEL

We consider a dynamic variant of the WTA problem, wherein the targets are visited sequentially by the bomber. Furthermore, we also incorporate feedback, in that the bomber is informed about the success/failure of a weapon upon deployment. This allows for dynamic decision making, where a decision is made as to whether a) an additional weapon is deployed on the current target or b) the bomber moves on to the next target in the sequence. In this regard, the decision rules for a hunter hunting targets whose reward takes values from a known (continuous valued) distribution is provided in Sato (1997). The scenario considered in this paper is simpler in that the target values are deterministic and known a priori.

Let $V(j,w)$ indicate the optimal cumulative reward ("pay-off" to go) that can be achieved when the bomber arrives at the j^{th} target with $w > 0$ weapons in hand. It follows that $V(j,w)$ must satisfy the Bellman recursion:

$$\begin{aligned} V(j,w) = \max_{u=0,1} \{&p_j(r_j + V(j+1,w-1)) \\ &+ q_j V(j,w-1), V(j+1,w)\}, \\ &j = 1,\ldots,(N-1), \end{aligned} \tag{6}$$

where the control action, $u = 0,1$ indicates whether the bomber should stay and deploy a weapon or move on to the next target. In (6), decision $u = 0$ results in the j^{th} target being destroyed with probability p_j and $u = 1$ results in the bomber moving on to the next target in the sequence. If target j is destroyed, the bomber receives an immediate reward of r_j. The optimal policy is therefore given by:

$$\begin{aligned} \mu(j,w) = \arg\max_{u=0,1} \{&p_j(r_j + V(j+1,w-1)) \\ &+ q_j V(j,w-1), V(j+1,w)\}, \\ &j = 1,\ldots,(N-1). \end{aligned} \tag{7}$$

If the bomber runs out of ammunition, there is no more reward to be gained i.e., the boundary condition is:

$$V(j,0) = 0, \; j = 1,\ldots,N, \tag{8}$$

Furthermore, if the bomber is at the N^{th} target and still has weapons at hand, the expected reward is given by:

$$V(N,w) = r_N(1 - q_N^w), \; w > 0. \tag{9}$$

In other words, q_N^w represents the probability that the N^{th} target is not destroyed by any of the w weapons. So, $1-q_N^w$ is the probability that it gets destroyed; thereby yielding the reward (9). The optimal policy can be obtained by the Backward Dynamic Programming (BDP) algorithm detailed below.

Algorithm BDP

1. Initialize $V(j,0) = 0, \; j = 1,\ldots,N$
2. Initialize $V(N,k) = r_N(1 - q_N^k), \; k = 1,\ldots,M$
3. Initialize $\mu(N,k) = 0, \; k = 1,\ldots,M$
4. for $j = (N-1)$ to 1
5.   for $k = 1$ to M
6.   Compute $V(j,k)$ as per (6)

If there was only 1 weapon left, we have from (6):

$$V(j,1) = \max_{u=0,1} \{p_j r_j, V(j+1,1)\}, \; j < N, \tag{10}$$

and $V(N,1) = p_N r_N$. So, a greedy policy is optimal and,

$$V(1,1) = \max_{j=1}^{N} p_j r_j. \tag{11}$$

The bomber, as expected, deploys the weapon on the target with the maximum expected reward. Note that **Algorithm** BDP has a time complexity of $\mathcal{O}(NM)$. However, for practical scenarios, one is unlikely to encounter large values of N or M, that make the algorithm computationally infeasible. In any case, it would be beneficial to understand the structure (if any) in the optimal policy. In the next section, we show that the optimal policy has a special structure. Indeed, it is a threshold policy (or a *control limit policy* - see Sec. 4.7.1 in Puterman (2005) for details). We show that a weapon from an inventory of k weapons is dropped on the j^{th} target iff k exceeds a threshold or control limit, $\kappa(j)$.

4. THRESHOLD POLICY

Let $\Delta_j(k) := V(j,k+1) - V(j,k)$ indicate the marginal reward yielded by assigning an additional weapon over and above k weapons to the downstream targets numbered j to N.

Proposition 1. $\Delta_j(k)$ is a monotonically decreasing function of k.

We shall prove the above proposition later. Notice however that the marginal reward yielded by the last (N^{th}) target in the sequence,

$$\Delta_N(k) = V(N,k+1) - V(N,k) = p_N r_N q_N^k, \tag{12}$$

is clearly a decreasing function of k given that $q_N < 1$. Suppose Proposition 1 is true i.e., $\Delta_{j+1}(k)$ is a monotonically decreasing function of k. Then, we can define $\kappa(j) = \min_{k=0,1,\ldots} k$ such that $p_j r_j \geq \Delta_{j+1}(k)$. The following result shows that a thresholding policy is optimal for the j^{th} target.

Lemma 2. If $\Delta_{j+1}(k)$ is a monotonically decreasing function of k,

$$\mu(j,k+1) = \begin{cases} 1, & k < \kappa(j), \\ 0, & \text{otherwise.} \end{cases}$$

Proof. From the Bellman recursion (6), we have: $V(j,k) \geq V(j+1,k)$. It follows that:

$$\begin{aligned} p_j(r_j + V(j+1,k)) + q_j V(j,k) &\geq p_j r_j + V(j+1,k) \\ &\geq V(j+1,k+1), \\ &\forall k \geq \kappa(j). \end{aligned} \tag{13}$$

where (13) follows from the definition of $\kappa(j)$. Recall the Bellman recursion (6):