Optimal Sequential Resource Allocation under Error-prone Success Assessment

K. Kalyanam[1], M. Pachter[2], D. Casbeer[3]

[1]*InfoSciTex Corporation, a DCS Company, Dayton, OH 45431, USA: krishnak@ucla.edu*

[2]*Air Force Institute of Technology, Wright-Patterson A.F.B., OH 45433, USA: meir.pachter@afit.edu*

[3]*Air Force Research Laboratory, Wright-Patterson A.F.B., OH 45433, USA: david.casbeer@us.af.mil*

Abstract We formulate a stochastic optimization problem, wherein finite resources are sequentially allocated to incoming tasks so as to maximize a cumulative reward. At each decision stage, the incoming task has known value. However, upon allocating a resource, a task is completed with probability less than one. Furthermore, the decision maker is informed about the task completion status via an error-prone feedback mechanism that is subject to both type I and type II classification errors. The decision maker may choose to reallocate a resource to the current task or move on to the next task in the sequence. We show that a resource is allocated if and only if the expected reward from a task exceeds a threshold. Furthermore, we establish a lower bound on the task completion probability under which the threshold is monotonic decreasing in the number of remaining resources.

1. Introduction

The operational scenario is the following. A Decision Maker (DM) with finite resources sequentially engages incoming tasks. Should the DM decide to allocate a resource to a task, it will be completed with probability $p < 1$. We assume that a single resource allocated to an incomplete task either leads to completion or it does not; there is no partial or cumulative task completion. Upon successful completion, the DM receives a known reward. However after allocating a resource, the DM only has access to an error-prone status report on the task. Indeed, the feedback process is an error prone classifier in that a completed task may be reported as being incomplete and vice-versa. Upon receiving the status report, the DM can either re-assign a resource to the current task or move on to the next task in the sequence. This decision obviously relies on the probability that the task is still not complete given the history of observations and the number of tasks and resources left. There is no turning back i.e., the DM is not allowed to revisit tasks that had been previously serviced. We are interested in the optimal feedback policy that results in the maximal expected cumulative reward. In particular, we are interested in the structural property, if any, of the optimal policy.

1.1. *Prior Work*

The framework we consider is applicable in military weapon target engagements, e.g., see Mastran and Thomas (1973), Kisi (1976), Aviv and Kress (1997), shooting problems, e.g., see Sato (1997a), Glazebrook and Washburn (2004), cybersecurity, e.g., see Hu et al. (2017), Gao et al. (2013), attacker-defender games, e.g., see Levitin and Hausken (2012) and more broadly to stochastic sequential resource allocation, e.g., see Sato (1996), Sato (1997b). In sequential assignment problems of the kind considered herein, the decision rule usually takes the form, *service the task iff its value is no less than a certain threshold c*, where the optimal c is to be determined. Moreover, one would expect the optimal threshold to be monotonic decreasing in the number of remaining resources. If the probability of successful task completion, $p = 1$, there is no need for repeated allocation to the same task and the resulting problem bears similarity to Revenue Management (RM) - wherein, the threshold monotonicity property is known to hold, e.g., see van Ryzin and Talluri (2005), Aydin et al. (2009). However, it is not obvious that the result holds for $p < 1$. To clearly distinguish this paper from our earlier work, we note that:

1) In Krishnamoorthy et al. (2016), a perfect information scenario is considered, where the DM is informed if the task was completed or not and the threshold monotonicity result has been established for this case.

2) In Krishnamoorthy et al. (2017), the rewards are considered to be random (as in RM) but known prior to decision time and the task status is perfectly known to the decision maker; the threshold monotonicity result has been established for this case as well. In addition, for this case, we also showed that the threshold is monotonic non-decreasing in the number of remaining tasks.

In this article, we extend the monotonicity result in Krishnamoorthy et al. (2016) to the case of an error-prone feedback mechanism. We also note that the result in Krishnamoorthy et al. (2016) does not follow directly from the result established in this work - see Remark 2 for alternate analysis.

In military parlance, our model embraces a Shoot-Look-Shoot (SLS) approach, in that homogenous resources are assigned one at a time with observations in between that assess the success or failure of the previous allocation. The analysis of SLS strategies in the presence of partial information was first reported in Aviv and Kress (1997). Gaver and Jacobs (1997) consider a conventional classifier model with type I and type II errors and provide effectiveness measures for different shooting tactics. A survey of the state of the art in SLS methods and the optimal use of information is provided in Glazebrook and Washburn (2004). It is also known that if additional complicating factors such as a search cost for finding a task or a scenario wherein resources can be replenished are considered, the monotonicity property breaks down, e.g., see Sato (1996), Sato (1997b).

In this article, we present a low resolution model of the SLS scenario where an imperfect binary classifier is employed. Our emphasis is on providing an analytical relationship between the probabilities of correctly classifying a completed task and misclassifying an incomplete task and the task completion probability under which the threshold monotonicity property holds.

2. Dynamic Model

Let the two possible observations after a resource is allocated to a task be given by $y = 1, 2$, where 1 indicates that the task is reported to be incomplete and 2 indicates that the task is reported to be complete. Furthermore, let the conditional probability that $y = 1$ given that the task is incomplete be given by q_1 and the conditional probability that $y = 1$ given that the task is complete be given by q_2.

2.1. *Monotonicity condition*

Let P_t be the prior probability that task t is incomplete. Now, suppose a resource is allocated to it. After expending the resource, the DM receives feedback/ observation: $y = 1, 2$, indicating that the task is reported to be incomplete and complete respectively. Let $P_t^+(y)$ indicate the a posteriori probability that task t is incomplete after receiving the report y. We have the conditional probabilities: $Prob(y = 1|incomplete) = q_1$ and $Prob(y = 1|complete) = q_2$.

THEOREM 1. *If $q_2 < q_1 \leq \frac{1}{1-p} q_2$, then $P_t^+(y) < P_t$, for all y.*

Proof. If $q_1 > q_2$, the probability of correctly identifying a completed task is more than the probability of mis-classifying an incomplete task. One would expect a reliable and "truthful" classifier to satisfy this condition. Let us refer to the probability that the task is incomplete prior to the observation as $\alpha_t = P_t(1-p)$. To compute the a posteriori probability, we use Bayes' theorem and get:

$$P_t^+(y=1) = \frac{q_1 \alpha_t}{q_1 \alpha_t + q_2 (1-\alpha_t)} \text{ and} \tag{1}$$

$$P_t^+(y=2) = \frac{(1-q_1)\alpha_t}{(1-q_1)\alpha_t + (1-q_2)(1-\alpha_t)}. \tag{2}$$

If $q_1 > q_2$, we have:

$$\begin{aligned} \beta = q_1 - q_2 > 0, \\ \Rightarrow \alpha_t \beta > \alpha_t^2 \beta, \text{ since } \alpha_t < 1, \\ \Rightarrow \alpha_t \beta + \alpha_t q_2 > \alpha_t^2 \beta + \alpha_t q_2, \\ \Rightarrow P_t^+(y=1) = \frac{\alpha_t q_1}{\alpha_t \beta + q_2} > \alpha_t. \end{aligned} \tag{3}$$

A similar argument shows that $P_t^+(y=2) < \alpha_t$. So, we have:

$$P_t^+(y=2) < \alpha_t < P_t^+(y=1). \tag{4}$$

In other words, given that the classifier is a reliable indicator, a negative report that the *task is incomplete* increases the posterior probability (compared to the prior) and a positive report has the opposite effect. In

addition, for monotonicity, we require:

$$\begin{aligned}
&P_t^+(y=1) \le P_t, \\
\Rightarrow &\frac{q_1(1-p)}{q_1 P_t(1-p) + q_2\left(1 - P_t(1-p)\right)} \le 1, \\
\Rightarrow &q_1(1-p)\left(1-P_t\right) \le q_2\left(1 - P_t + (1-p)\right), \\
\Rightarrow &\left(q_1 - q_2\right)(1-p)\left(1-P_t\right) \le q_2 p. \qquad (5)
\end{aligned}$$

Since $P_t \le 1$, a sufficient condition that guarantees monotonicity is given by:

$$p \ge 1 - \frac{q_2}{q_1}. \qquad (6)$$

To summarize, if $q_1 > q_2$ and the successful task completion probability p is sufficiently high (6), the a posteriori probabilities satisfy: $P_t^+(y) \le P_t$, $y = 1,2$.

□

REMARK 1. *Note that $q_1 = q_2$ is a degenerate case, since the classifier is redundant and renders no useful information about the outcome of the allocation. Indeed, Bayes' update tells us that, for this case, $P_t^+(y = 1,2) = \alpha_t$ i.e., the posterior and prior probabilities are the same. Also, $q_1 < q_2$ is the strange case of a counter-indicator (or a reliable "liar") and so the conditions for monotonicity are the same as established before, except that we switch q_1 for q_2 and vice-versa.*

2.2. *Stochastic Dynamic Programming*

Let there be N tasks waiting to be serviced. We use $t = 1,\ldots,N$ to indicate the current task index. If task t is completed, the DM will receive a running reward of r_t. Suppose the DM is currently facing task t with k resources in hand. Further suppose that the probability that task t is incomplete be given by $P_t \le 1$. If the DM just received task t, we would of course have $P_t = 1$. Let $V(t,k|P_t)$ indicate the optimal expected cumulative reward i.e., "payoff to go", that can be achieved thereafter. It follows that $V(t,k|P)$ must satisfy the Bellman recursion:

$$V(t,k|P_t) = \max_{u=0,1}\left\{u\left(pP_t r_t + \sum_y Prob(y)V(t,k-1|P_t^+(y))\right) + (1-u)V(t+1,k|1)\right\}, \qquad (7)$$

where the control action, $u = 1$ indicates that an additional resource is to be spent on the current task and $u = 0$ indicates that the DM will move on to the next task in the sequence. If the decision $u = 1$ is taken, the immediate expected reward of $pP_t r_t$ is gained. After each allocation, the DM receives the status report, y. In (7), $Prob(y)$ is the probability that report y is generated when $u = 1$ is chosen. Indeed, we have:

$$Prob(y=1) = q_1 P_t(1-p) + q_2\left(1 - P_t(1-p)\right) \text{ and} \qquad (8)$$

$$Prob(y=2) = (1-q_1)P_t(1-p) + (1-q_2)\left(1 - P_t(1-p)\right). \qquad (9)$$

The a posteriori probability that the task is still incomplete after receiving y, is given by $P_t^+(y)$ and is computed, as before, using Bayes' theorem:

$$P_t^+(y=1) = \frac{q_1 P_t(1-p)}{Prob(y=1)} \text{ and} \qquad (10)$$

$$P_t^+(y=2) = \frac{(1-q_1)P_t(1-p)}{Prob(y=2)}. \qquad (11)$$

If the DM faces the last task and still has $k > 0$ resources at hand, the expected reward is given by:

$$V(N,k|1) = r_N(1-q^k), \qquad (12)$$

where $q = 1 - p$. In other words, q^k represents the probability that the last task is not completed by any of the k resources. So, $1 - q^k$ is the probability that it gets completed; thereby yielding the reward (12).

The optimal feedback policy is therefore given by:

$$\mu(t,k|P_t) = \arg\max_{u=0,1} \left\{ u\left(pP_t r_t + \sum_y Prob(y) V(t,k-1|P_t^+(y)) \right) + (1-u)V(t+1,k|1) \right\}. \tag{13}$$

The solution entails solving the Stochastic Dynamic Program (SDP) given by (7). Here, the stage $t = 1,\ldots,N$ and the state is (k,P_t), where k is the number of resources in hand at stage t.

2.3. *Dynamic Programming Solution*

We note that the Dynamic Programming (DP) equation (7) is unconventional, in that the variable P_t takes continuous values in $(0,1]$. So, standard solution methods such as value/ policy iteration do not directly apply. However, we point out that for any finite instance of the problem with say M resources and N tasks, one can pre-compute all possible values that the posterior probability, $P_t^+(y_1,\ldots,y_n)$ can take as a function of the observation history $y_1,\ldots,y_n$ over $n \leq M$ resources allocated to task t. Indeed, let the one-step Bayes update (10) and (11) be denoted by $P_t^+(y=1) = \mathscr{F}_1(P_t)$ and $P_t^+(y=2) = \mathscr{F}_2(P_t)$ respectively. Let us define the indexed set,

$$\begin{aligned} &\mathscr{P}_n = \cup_{i=1,2} \cup_{P\in\mathscr{P}_{n-1}} \mathscr{F}_i(P),\ n = 1,\ldots,(M-1), \\ &\mathscr{P}_0 = \{1\}. \end{aligned} \tag{14}$$

It is clear that the cardinality of $\mathscr{P}_n$ is 2^n. Having precomputed the sets $\mathscr{P}_n$, $\forall n < M$, the DP equation (7) can be restated as follows:

$$\begin{aligned} V(t,k|P) = \max_{u=0,1} \left\{ u\left(pPr_t + \sum_{i=1}^{2} Prob(i) V(t,k-1|\mathscr{F}_i(P)) \right) + (1-u)V(t+1,k|1) \right\}, \\ k = 1,\ldots,M, \forall P \in \cup_{n=0}^{M-k} \mathscr{P}_n. \end{aligned} \tag{15}$$

So, $V(t,k|P)$ can be computed so long as $V(t,k-1|\mathscr{F}_i(P))$ and $V(t+1,k|1)$ are available. However, $V(t,1|P)$ depends only on $V(t+1,1|1)$ since $V(t,0|\mathscr{F}_i(P)) = 0$. For $k > 1$, we have, $\mathscr{F}_i(P) \in \cup_{n=1}^{M-k+1} \mathscr{P}_n$ since $P \in \cup_{n=0}^{M-k} \mathscr{P}_n$. So, one could solve (15) using backward Dynamic Programming. Indeed, we first compute:

$$V(N,k|1) = r_N(1-q^k),\ k = 1,\ldots,M. \tag{16}$$

We then proceed to compute $V(t,k|P)$, $\forall P \in \mathscr{P}_{M-k}$ in the decreasing order of t from $N-1$ to 1 and increasing order of k from 1 to M using (15). In essence, the state component P is quantized and as such the DP algorithm deals with a finite discrete state space. However, scalability is an issue since the total number of states is $\mathscr{O}(N \times M \times 2^M)$.

Although interesting in its own right, we shall not dwell any longer on the computational aspects of the DP algorithm. Instead, our focus here is to characterize the optimal solution. As mentioned earlier, under fairly reasonable assumptions, the optimal policy can be shown to have a special structure. Indeed, a resource from an available inventory of k resources is allocated to a task of value r_t iff $P_t pr_t$ exceeds a threshold. In the next section, we prove that a threshold policy is optimal and furthermore, the threshold is monotonic decreasing in the inventory level.

3. Monotone Threshold Policy

Let $\Delta_{t+1}(k) := V(t+1,k|1) - V(t+1,k-1|1)$ indicate the expected marginal reward yielded by assigning an additional resource over and above $k-1$ resources to the future tasks $t+1$ to N.

PROPOSITION 1. *$\Delta_t(k)$ is a monotonically decreasing function of k.*

We shall prove the above proposition later. Notice however that the marginal reward yielded by the last task in the sequence,

$$\Delta_N(k) = V(N,k|1) - V(N,k-1|1) = pr_N q^{k-1}, \tag{17}$$

is clearly a decreasing function of k given that $q < 1$.

Suppose Proposition 1 is true i.e., $\Delta_{t+1}(k)$ is a monotonically decreasing function of k. For any $P \in (0,1]$, we

can define $\kappa_t(P) = \min_{k=1,2,\cdots} k$ such that $pPr_t \geq \Delta_{t+1}(k)$. Indeed, $\kappa_t(P)$ is the smallest positive integer such that pPr_t is no smaller than the marginal reward $\Delta_{t+1}(k)$. The following result shows that a thresholding policy is optimal.

LEMMA 1. *If $\Delta_{t+1}(k)$ is a monotonically decreasing function of k, the optimal policy:*

$$\mu(t,k|P_t) = \begin{cases} 0, \ k < \kappa_t(P_t), \\ 1, \ \textit{otherwise.} \end{cases}$$

Proof. From the Bellman recursion (7), we have: $V(t,k|P_t) \geq V(t+1,k|1)$. It follows that:

$$\begin{aligned} pP_t r_t + \sum_y Prob(y) V(t,k-1|P_t^+(y)) &\geq pP_t r_t + V(t+1,k-1|1) \\ &\geq V(t+1,k|1), \forall k \geq \kappa_t(P_t), \end{aligned} \tag{18}$$

where (18) follows from the definition of $\kappa_t(P_t)$. Recall the Bellman recursion (7):

$$V(t,k|P_t) = \max_{u=0,1} \left\{ u \left(pP_t r_t + \sum_y Prob(y) V(t,k-1|P_t^+(y)) \right) + (1-u) V(t+1,k|1) \right\},$$

$$\Rightarrow V(t,k|P_t) = pP_t r_t + \sum_y Prob(y) V(t,k-1|P_t^+(y)), \ k \geq \kappa_t(P_t). \tag{19}$$

$$\Rightarrow \mu(t,k|P_t) = 1, \ \forall k \geq \kappa_t(P_t). \tag{20}$$

We shall prove the second part of the result, i.e., $\mu(t,k|P_t) = 0, \ k < \kappa_t(P_t)$ by induction on k. Recall the definition of $\kappa_t(P_t)$, which gives us:

$$pP_t r_t + V(t+1,k-1|1) < V(t+1,k|1), \ k < \kappa_t(P_t). \tag{21}$$

If $\kappa_t(P_t) = 1$, there is nothing left to prove. So, suppose $\kappa_t(P_t) > 1$. From the Bellman recursion (7), we have:

$$\begin{aligned} V(t,1|P_t)) &= \max_{u=0,1} \{ upP_t r_t + (1-u) V(t+1,1|1) \} \\ &= V(t+1,1|1), \end{aligned} \tag{22}$$

where (22) follows from (21) since $\kappa_t(P_t) > 1$. Suppose $V(t,h-1|P_t) = V(t+1,h-1|1)$ for some $h < \kappa_t(P_t)$. We have the following monotonicity result (for proof, see Sec 2.1):

$$\begin{aligned} q_2 < q_1 \leq \frac{1}{1-p} q_2 \\ \Rightarrow P_t^+(y) \leq P_t, \ y = 1,2. \end{aligned} \tag{23}$$

Now $V(t,h-1|P_t) = V(t+1,h-1|1)$ implies that it is optimal to engage the next task in the sequence if the probability that task t is incomplete is P_t and the inventory level is $h-1$. It follows that if the probability that t is incomplete is even lower, the DM must move on to the next task in the sequence and so, $V(t,h-1|P_t^+(y)) = V(t+1,h-1|1), \ y = 1,2$. Using the induction argument, the Bellman recursion (7) yields:

$$\begin{aligned} V(t,h|P_t) &= \max_{u=0,1} \left\{ u \left(pP_t r_t + \sum_y Prob(y) V(t,h-1|P_t^+(y)) \right) + (1-u) V(t+1,h|1) \right\}, \\ &= \max_{u=0,1} \{ u(pP_t r_t + V(t+1,h-1|1)) + (1-u) V(t+1,h|1) \}, \\ &= V(t+1,h|1), \end{aligned} \tag{24}$$

where (24) follows from (21) since $h < \kappa_t(P_t)$. Combining (22) and (24), we have:

$$V(t,k|P) = V(t+1,k|1), \ \forall k < \kappa_t(P), \tag{25}$$

$$\Rightarrow \mu(t,k|P) = \begin{cases} 0, \ \forall k < \kappa_t(P). \\ 1, \ \text{otherwise.} \end{cases}$$

□

The above result tells us that 1 out of the current inventory of k resources is deployed on the current task iff the immediate expected reward, $pP_t r_t$ is no less than the marginal reward obtained by assigning an additional resource over and above $k-1$ resources to future tasks. So, a threshold policy is optimal, with the threshold value given by $\Delta_{t+1}(k)$.

REMARK 2. *The special case of perfect feedback information is recovered by setting $q_1 = 1$ and $q_2 = 0$. However, Lemma 1 relies on the condition $(1-p)q_1 \leq q_2$ and therefore the above result does not hold for the perfect information case. A separate analysis is required for this case - see Krishnamoorthy et al. (2016).*

THEOREM 2. *$\Delta_t(k)$ is monotonically decreasing in the inventory level k.*

Proof. We prove the result by induction on the task index, t. From (17), we know that $\Delta_N(k)$ is monotonic decreasing in k. Let us suppose that $\Delta_{t+1}(k)$ is also a decreasing function of k. Combining (19) and (25), the optimal expected cumulative reward takes the form:

$$V(t,k|P_t) = \begin{cases} V(t+1,k|1),\ k < \kappa_t(P_t) \\ pP_t r_t + V(t+1,k-1|1),\ k = \kappa_t(P_t) \\ pP_t r_t + \sum_y Prob(y) V(t,k-1|P_t^+(y)),\ k > \kappa_t(P_t). \end{cases} \tag{26}$$

By repeated application of (26), we have for $k > \kappa_t(P_t)$:

$$\begin{aligned} V(t,k|P_t) &= pP_t r_t + \sum_y Prob(y) \left[pP_t^+(y) r_t + \sum_y \dots \right] \\ &= pP_t r_t + pr_t \sum_y Prob(y) P_t^+(y) + \sum_y Prob(y) \left[\sum_y \dots \right]. \end{aligned} \tag{27}$$

However, from the Bayes' update (10) and (11), we note that

$$\sum_y Prob(y) P_t^+(y) = P_t(1-p) = P_t q. \tag{28}$$

So, we can write:

$$\begin{aligned} V(t,k|P_t) &= pP_t r_t + pP_t r_t q + \sum_y Prob(y) \left[\sum_y \dots \right], \\ \Rightarrow V(t,k|P_t) &= pP_t r_t \sum_{i=0}^{k-\kappa_t(P_t)} q^i + V(t+1, \kappa_t(P_t)-1|1),\ k \geq \kappa_t(P_t). \end{aligned} \tag{29}$$

So, we have the marginal reward given by:

$$\begin{aligned} \Delta_t(k) &= V(t,k|1) - V(t,k-1|1) \\ &= \begin{cases} \Delta_{t+1}(k),\ k < \kappa_t(1), \\ pr_t q^{k-\kappa_t(1)},\ k \geq \kappa_t(1). \end{cases} \end{aligned} \tag{30}$$

We proceed to show that $\Delta_t(k)$ as prescribed by (30) is a decreasing function of k. By our induction argument, $\Delta_t(k)$ decreases as k goes from 0 to $\kappa_t(1)-1$. From the definition of $\kappa_t(1)$, we have:

$$pr_t < \Delta_{t+1}(\kappa_t(1)-1).$$

For $k \geq \kappa_t(1)$, $\Delta_t(k) = pr_t q^{k-\kappa_t(1)}$ is a decreasing function of k since $q < 1$. □

REMARK 3. *Note that, in addition, if the tasks arrive in the order of increasing value, i.e., $r_1 \leq r_2 \leq \dots \leq r_N$, it can be shown that the threshold is also monotonic non-increasing in t. In particular, if all rewards are the same, at least one resource will be allocated to each task.*

4. Conclusion

We considered a dynamic resource allocation problem, wherein arriving tasks are sequentially serviced by a DM equipped with homogenous resources. Feedback on the task completion status upon allocation is available

in the form of an error-prone status report. Stochastic Dynamic Programming yields the optimal policy which specifies that a resource is allocated to a task iff the immediate expected reward exceeds a threshold. Under mild conditions on the feedback classifier error probabilities, we show that the threshold is monotonic decreasing in the number of resources. The work is readily applicable to a dynamic weapon target assignment scenario, where an error prone Battle Damage Assessment (BDA) report is available.

References

Aviv, Y., Kress, M., (1997). Evaluating the effectiveness of shoot-look-shoot tactics in the presence of incomplete damage information. *Military Operations Research* 3 (1), 79–89.

Aydin, S., Akçay, Y., Karaesmen, F., (2009). On the structural properties of a discrete-time single product revenue management problem. *Operations Research Letters* 37 (4), 273–279.

Gao, X., Zhong, W., Mei, S., (2013). A game-theory approach to configuration of detection software with decision errors. *Reliability Engineering and System Safety* 119, 35–43.

Gaver, D. P., Jacobs, P. A., (1997). Probability models for battle damage assessment (simple shoot-look-shoot and beyond). Tech. Report NPS-OR-97-014, Naval Postgraduate School, Monterey, CA.

Glazebrook, K. D., Washburn, A., (2004). Shoot-look-shoot: A review and extension. *Operations Research* 52 (3), 454–463.

Hu, X., Xu, M., Xu, S., Zhao, P., (2017). Multiple cyber attacks against a target with observation errors and dependent outcomes: Characterization and optimization. *Reliability Engineering and System Safety* 159, 119–133.

Kisi, T., (1976). Suboptimal decision rule for attacking targets of opportunity. *Naval Research Logistics* 23 (3), 525–533.

Krishnamoorthy, K., Casbeer, D., Pachter, M., (2017). Monotone optimal threshold feedback policy for sequential weapon target assignment. *AIAA Journal of Aerospace Information Systems* 14 (1), 68–72.

Krishnamoorthy, K., Rathinam, S., Casbeer, D., Pachter, M., (2016). Optimal threshold policy for sequential weapon target assignment. *In IFAC Symposium on Automatic Control in Aerospace.* Vol. 49 of IFAC-PapersOnLine. Sherbrooke, Quebec, Canada, pp. 7–10.

Levitin, G., Hausken, K., (2012). Resource distribution in multiple attacks with imperfect detection of the attack outcome. *Risk Analysis* 32 (2), 304–318.

Mastran, D. V., Thomas, C. J., (1973). Decision rules for attacking targets of opportunity. *Naval Research Logistics* 20 (4), 661–672.

Sato, M., (1996). A sequential allocation problem with search cost where the shoot-look-shoot policy is employed. *Journal of the Operations Research Society of Japan* 39 (3), 435–454.

Sato, M., (1997a). On optimal ammunition usage when hunting fleeing targets. *Probability in the Engineering and Informational Sciences* 11 (1), 49–64.

Sato, M., (1997b). A stochastic sequential allocation problem where the resources can be replenished. *Journal of the Operations Research Society of Japan* 40 (2), 206–219.

van Ryzin, G. J., Talluri, K. T., (2005). An Introduction to Revenue Management. *INFORMS TutORials in Operations Research*. Springer Berlin, Ch. 6, pp. 142–194.