Proceedings of the 2015 Winter Simulation Conference
L. Yilmaz, W. K. V. Chan, I. Moon, T. M. K. Roeder, C. Macal, and M. D. Rossetti, eds.

SIMULATION-DRIVEN TASK PRIORITIZATION USING A RESTLESS BANDIT MODEL FOR ACTIVE SONAR MISSIONS

Cherry Y. Wakayama

Maritime Systems Division
SPAWAR Systems Center Pacific
San Diego, CA 92152, USA

Zelda B. Zabinsky

Industrial & Systems Engineering Department
University of Washington
Seattle, WA 98195, USA

ABSTRACT

We consider a task prioritization problem of an active sonar tracking system when available ping resources may not be sufficient to sustain all tracking tasks at any particular time. In this problem, the time-varying conditions of a tracking task are represented by a finite-state discrete-time Markov decision process. The objective is to find a policy which decides at each time interval which tracking tasks to perform so as to maximize the aggregate reward over time. This paper addresses the derivation of the Markov chain parameters from the sonar tracking system simulations, the establishment of task prioritization as a restless bandit (TPRB) problem, and the TPRB policy obtained by a primal-dual index heuristic based on a first-order linear programming relaxation to the TPRB problem. The superior performance of the resulting TPRB policy is demonstrated using Monte Carlo simulations on various multi-target scenarios.

1 INTRODUCTION

Active sonar systems detect underwater objects of interest by transmitting acoustic waveforms, or pings, from sources and detecting the echoes with receivers. Active sonar systems are especially employed in military applications, and can carry out a variety of tasks, such as surveillance and multi-target tracking (Grimmett and Coraluppi 2006). This paper describes an active sonar system composed of spatially distributed sonar nodes or co-located source/receiver pairs for multi-target tracking as illustrated in Figure 1. Since the sonar tracking system contains numerous transmit/receive nodes, and the number of simultaneous transmits from multiple sources is limited due to potential interference and communications and processing constraints, effective sonar resource management is required to use the system capabilities efficiently. There are generally two related aspects of resource management for a sonar tracking system: scheduling and task prioritization. Scheduling decides how to commit resources (sources, waveforms and time slots) between a given set of tasks so as to best accomplish the tasks. An adaptive ping scheduling approach for active sonar systems for multi-objective anti-submarine warfare (ASW) area search and tracking missions is developed by Wakayama et al. (2015). However, in certain operating conditions, there may not be enough resources to perform all the required tasks without compromising quality, e.g., the number of tracking tasks on hand is greater than the number of orthogonal waveforms that the system can transmit simultaneously without interference. Task prioritization determines the set of active tasks to focus on in such instances, and becomes critical to the performance of the sonar resource management and to subsequent mission success. In this paper, we study the dynamic task prioritization problem associated with the allocation of pings to tracking tasks over time.

The dynamic task prioritization problem fits naturally into the framework of multi-armed bandits (Mahajan and Teneketzis 2008). In this problem, tracking tasks may be viewed as bandits. At each ping interval, ping(s) are allocated to selected bandit(s) to generate reward. In particular, this problem is characterized as a *restless* bandit problem because track quality deteriorates when the tracking task is not

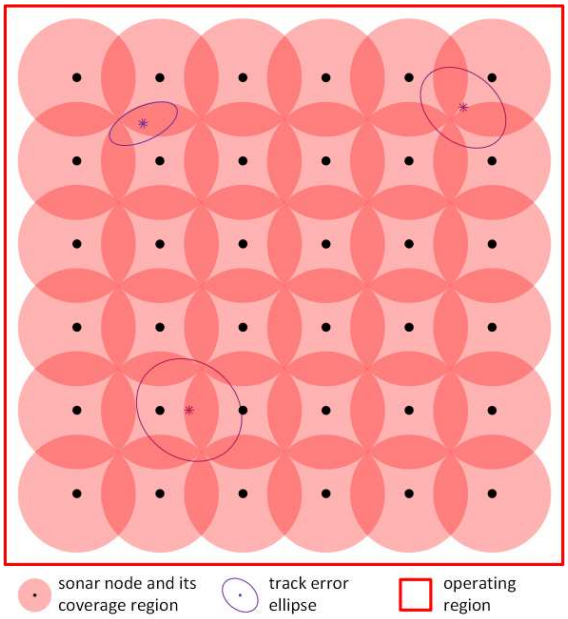


Figure 1: Multi-target tracking scenario for an active sonar system.

being processed (Whittle 1998). With the assumption of the utilization of a multi-target tracker in the sonar tracking system, we define a tracking task as a target which is previously detected and associated in the tracker. The tracking task states represent the quantized areas of uncertainty (AOU) owing to track quality during the track updates in the tracker. We assume that the tracking task states are perfectly observable with the utilization of a tracker. We also assume that the tracking task states evolve independently of each other. At each instance of time or ping interval, tasks are selected to which pings must be allocated, and the sonar tracking system collects a reward as a function of the states. When a track is selected for prioritization (i.e., a ping is allocated for the selected track), information on its position and velocity is obtained if the ping results in a detection, and hence it may result in an improved AOU state (smaller AOU results in faster target acquisition time). For the tracks that are not selected, information is being lost, because the targets will certainly be performing evasive maneuvering. The tracking tasks continue to change states whether they are being selected or not. Each tracking task has potentially different state space, and the state transition rules for each task with each prioritization decision will be different in general. Considering complexity in active sonar tracking systems, the Markovian transition rules with respect to the binary prioritization decisions on a tracking task are derived from sonar tracking system simulations.

The complex dynamics of a sonar tracking system are evident from maneuvering targets, and imperfect and noisy sensors whose detection probability is dependent on acoustic environment conditions, target states and waveform characteristics. In this instance, simulation is very effective in modeling and capturing dynamic effects of a sonar tracking system. Monte Carlo track samples are generated and the statistics about the AOU state evolutions associated with a tracking task are obtained by analyzing the resulting simulated data sets. Extracting efficient high-level models from complex dynamic processes is highly desirable for real-time control applications such as dynamic task prioritization for sonar resource management.

The estimated Markov transition matrices are utilized to formulate the task prioritization as a restless bandit (TPRB) problem. The TPRB problem is then solved using a primal-dual heuristic index algorithm based on a first-order linear programming relaxation (Bertsimas and Niño-Mora 2000). The approach to solving the TPRB problem is referred to as the *TPRB scheme* in this paper. The TPRB scheme provides a feasible policy with a guarantee for its suboptimality, and has been demonstrated numerically to have excellent performance. The tracker provides updates on tracking tasks at regular intervals, and the TPRB

problem is solved repeatedly with an updated initial condition to derive task prioritization policy throughout the mission duration.

This paper proceeds as follows. Section 2 describes a general active sonar tracking system model and its submodules focusing on tracking tasks. Section 3 discusses the modeling of individual tracking task dynamics using a finite-state discrete-time Markov chain from the simulated data sets. Section 4 presents a restless bandit model formulation for the dynamic task prioritization problem. Section 5 explains a solution approach to the TPRB problem based on a first-order linear programming relaxation. Section 6 compares the performance of the resulting TPRB policy with the conventional policies including a round robin policy and a greedy policy on various multi-target scenarios, and Section 7 provides insights and conclusions.

2 SYSTEM MODEL

A general active sonar tracking system model is shown in Figure 2, and is used for simulation and analyzing the task prioritization techniques. In this closed-loop architecture, information on targets is obtained by active sonar measurements and processed by the multi-target tracker and the resource management modules (task prioritization and scheduling) to derive effective ping decisions, and the ping decisions are then fed back to the sonar system for improved tracking. In this section, every module of the system model is briefly described.

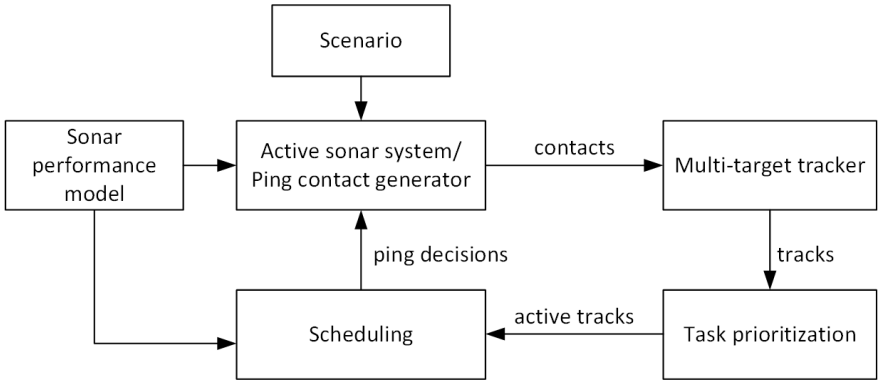


Figure 2: Active sonar tracking system model.

Scenario

A field of multiple sensor nodes is positioned in an acoustic environment of interest and its operating region is identified. In this study, the scenario is formed with tracking tasks of N targets. The target tracks are sampled from a set of randomly-generated tracks. Tracks are generated with random headings (uniform between -180° and 180°), random speeds (uniform between minimum and maximum assumed speeds) and random initial positions (uniform within the operating region). A discrete-time, nearly-constant-velocity model with assumed motion noise is used to update each track for a specified scenario time window.

Sonar Performance Model

The sonar performance model computes mean signal-to-noise ratios (SNR) or signal excesses (SE) for a given sensor field configuration consisting of sources, receivers and moving targets, and is an essential element of the ping decision derivation and the sonar system simulation. Detection probability predictions can then be computed from the SE values. For this analysis, mean SE values are obtained using Signal Systems Corporation's Multistatic Sonar Performance Model software (MSPM 2010).

Ping Contact Generator

A contact generator is required in order to test and evaluate sensor management techniques. This module provides target contacts for each receiver, given a waveform transmission from a particular source. Target contacts are derived from simulated tracks, and contain information including SNR, bearing, arrival time for ranging, and range-rate if available. They are modeled by obtaining mean SE from the sonar performance model, and adding a random fluctuation term from an assumed Gaussian distribution, along with assumed measurement (bearing, time, and range-rate) errors. Target contacts are generated when the resulting SE is positive. For this study on track prioritization scenarios, it is assumed that false contacts have already been removed by the multi-target tracker and hence they are not included in the ping contact generator.

Multi-target Tracker

The purpose of the tracker is to estimate the positions and velocities of targets of interest from the detection contacts. This process requires the removal of false contacts, the association of true contacts, and the fusion of contact information. There is a large literature on sensor data fusion and target tracking (Bar-Shalom et al. 2011). An appropriate tracker implementation must balance the tradeoff between complexity and quality depending on the applications. In this paper, a simple Kalman filter based tracker is implemented. The track outputs of the tracker are represented by their expected 2-dimensional positional and velocity state vectors and covariance matrices. Their AOU states are calculated from the 2-dimensional positional covariance matrix.

Task Prioritization

The task prioritization module dynamically determines the set of active tracks to focus on at any particular ping interval in order to optimize the overall sonar system performance. Task prioritization becomes critical as the number of tracking tasks on hand becomes higher and available resources may not be sufficient to sustain all tracks. The number of active tracking tasks that can be scheduled at each ping time interval is constrained by the number of orthorgonal waveforms available for the sonar system as well as processing and communications limitations. The main focus of this paper is dynamic task prioritization, and an optimization problem formulation for the task prioritization problem is discussed in detail in Section 4.

Scheduling

The scheduling module together with the task prioritization module control the performance of a sonar tracking system. Given the active tracking tasks selected by the task prioritization module at each ping time interval, the scheduling module provides the ping decisions on which source/waveform pairs are best to ping next. The performance is measured by the instantaneous detection probability and the resulting AOU of each active track. The scheduling strategy is to choose a source/waveform pair for each active tracking task that will result in a minimum AOU and a detection probability of greater than 0.5.

3 TRACKING TASK DYNAMIC MODEL FROM SYSTEM SIMULATION

We seek to predict the next AOUs of a given tracking task with respect to the binary prioritization decisions on the task using a Markov chain. Since it is simply not feasible to obtain real-world data, the AOU data required for Markov chain construction are generated from Monte Carlo track samples using the sonar tracking system model (Section 2). We then apply Markov chain calculations to each of the simulated data sets corresponding to each of the prioritization decisions to obtain a finite transition probability matrix with the states representing the quantized AOUs. These Markov chains represent a priori statistics about the tracking task dynamic processes for the task prioritization optimization problem discussed in Section 4.

3.1 AOU Data Set Generation

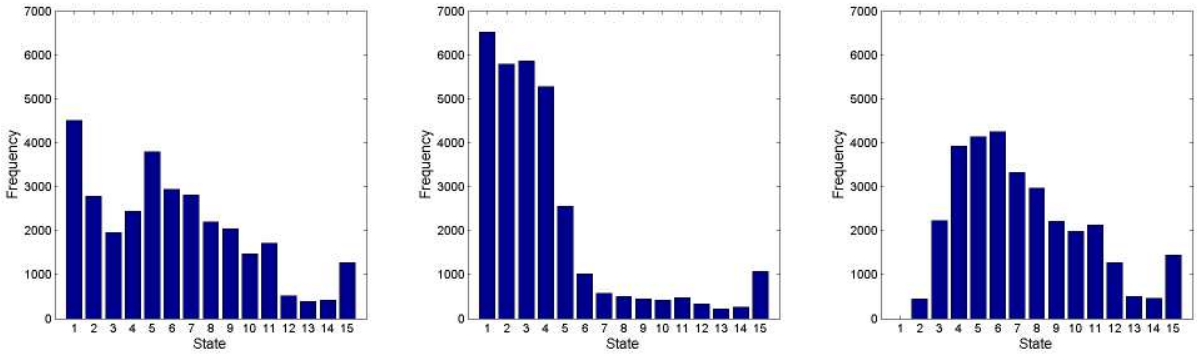
The sonar tracking system simulation framework for AOU data generation consists of a discrete event simulation model with an embedded optimization module or scheduling module, a sonar performance model, a ping contact generator and a multi-target tracker (Section 2). The user can modify sensor configurations, environment conditions and simulation parameters in the scenario module. A specified number of Monte Carlo tracks are generated with random headings, speeds and initial positions, and assumed motion noise. For each sampled track at each ping time interval, two data pairs, (δ, δ^1) and (δ, δ^0), are obtained with binary track prioritization decisions, namely active and passive (we identify superscript 1 with active (i.e., the track is selected and a ping is allocated for the track), and superscript 0 with passive (i.e., the track is not selected and is not considered in the ping decision generation)). The AOU of the sampled track just before the prioritization decision is denoted by δ, and calculated from the current predicted positional covariance matrix. With the active decision on the sampled track at any given interval, the scheduling module provides a ping (source/waveform pair) optimized for the track, the ping contact generator provides a target contact (if any) and the multi-target tracker updates its position and velocity estimates from the contact measurement. The active AOU, δ^1, is then calculated from the updated positional covariance matrix. With the passive decision, no contact is generated. The multi-target tracker propagates its position and velocity estimates and the passive AOU, δ^0, is calculated. Typically, the AOU grows rapidly in time without successive measurements. Grouping those data pairs according to the active or passive decision results in two data sets, denoted by D^1 and D^0 respectively.

3.2 Markov Transition Matrices

We calculate a state transition matrix from each data set, thereby obtaining two transition matrices denoted by $P^1$and P^0 corresponding to the data sets D^1 and D^0 respectively. A key step when creating a Markov chain is determining the appropriate quantization of states. To identify the number and quantization intervals, we analyze the data as well as consider the operational requirements of the sonar tracking system. In some tracking applications, it may be desirable to maintain the AOUs of the targets of interest below a certain threshold as dictated by the system requirements. When the AOU becomes greater due to successive misses, a track may be considered lost. In this paper, the quantization of AOU states is done by analyzing detection patterns and resulting AOUs. Histograms are useful in ensuring there are enough data points available to capture the state transitional behavior and establishing the quantization intervals or bin ranges. Once the intervals are created, each of the data pairs are labeled with the intervals to which they correspond. After the data pairs are labeled, the number of transitions from one interval to the next is counted and empirical state transition probabilities are calculated yielding the transition matrices P^1 and P^0. The state quantization for the examples in this paper is shown in Table 1. Figure 3 includes three histograms illustrating the distributions of the initial and next AOU states obtained from the simulated data sets. Figure 3a corresponds to the values of δ, and Figures 3b and 3c correspond to the values of δ^1 and δ^0 respectively. It is observed that with the active decision the mass of the distribution shifted towards the smaller AOU states and with the passive decision towards the larger AOU states.

Table 1: State Quantization

States	AOU ranges (km^2)	States	AOU ranges (km^2)	States	AOU ranges (km^2)
1	(0,0.35]	6	(1.8,3]	11	(17,27]
2	(0.35,0.5]	7	(3,5]	12	(27,40]
3	(0.5,0.7]	8	(5,8]	13	(40,62]
4	(0.7,1]	9	(8,12]	14	(62,100]
5	(1,1.8]	10	(12,17]	15	> 100



(a) (b) (c)

Figure 3: Histograms of the data sets used in the simulation examples: (a) initial AOU data $\{\delta\}$, (b) next AOU data with the active decision $\{\delta^1\}$ and (c) next AOU data with the passive decision $\{\delta^0\}$.

4 RESTLESS BANDIT FORMULATION

In this section, we provide a brief description of key features of the classical multi-armed bandit (MAB) problem. We then formulate the task prioritization as a variant of the classical MAB problem, known as restless bandit problem, and refer to it as the TPRB problem.

Multi-armed bandit problems are a class of sequential resource allocation problems concerned with allocating one or more resources among several competing projects (Mahajan and Teneketzis 2008). In the classical MAB problem at each instant of time a single resource is allocated to one of many competing projects. The project to which the resource is allocated changes its state according to a random dynamical process with known statistical descriptions and generates a reward. The remaining projects remain frozen and contribute no reward. Projects are assumed independent. Gittins (1979) showed that an optimal solution is of the index type. The Gittins indices can be computed separately for each project (or bandit) and thus, finding an optimal policy reduces to selecting the project with the highest index.

The task prioritization problem can be seen as a variation of the classical MAB problem. In the following we introduce the notation and provide the formal problem definition. We consider a collection of N tracking tasks (bandits), labeled $n \in \mathcal{N} = \{1,\ldots,N\}$. At ping interval k, task n can be in one of the finite number of states $i_n^k \in \mathcal{S}_n$. Let $a_n^k \in \{0,1\}$ be the action for task n at ping time k. If task n in state i_n^k is selected (i.e., $a_n^k = 1$), then an active reward $R^1_{i_n^k}$ is earned, and its state changes in a Markovian fashion, according to an active transition probability matrix P^1 into state j_n^{k+1} with probability $p^1_{i_n^k j_n^{k+1}}$. If the task is not selected (i.e., $a_n^k = 0$), then a passive reward $R^0_{i_n^k}$ is received, and its state then changes according to a passive transition probability matrix P^0 into state j_n^{k+1} with probability $p^0_{i_n^k j_n^{k+1}}$. The tracking tasks are characterized as restless bandits because each track's AOU state evolves over time even when it is not being selected (Whittle 1998).

In the TPRB problem, the system reward represents the value assigned according to the track quality (smaller AOU states correspond to higher track quality). If task n is selected or active, information may be updated depending on whether the ping will result in a detection or not. This probability is captured by P^1. When the state of the n-th task is i_n^k, we suppose that the active reward is the mean track quality value, which can be represented as

$$R^1_{i_n} = \sum_{j_n \in \mathcal{S}_n} p^1_{i_n j_n} V_{i_n}, \tag{1}$$

where V_{i_n} is the track quality value assigned for being in state i_n, which is pre-specified by the system operator. Similarly, the passive reward is defined as

$$R^0_{i_n} = \sum_{j_n \in \mathscr{S}_n} p^0_{i_n j_n} V_{i_n}. \tag{2}$$

In this formulation, rewards are time discounted by a discount factor β, where $0 < \beta < 1$.

At each ping interval $k = 0, 1, \ldots, K$, exactly M tasks must be selected, i.e., $\sum_{n=1}^{N} a^k_n = M$, where $1 \leq M < N$, and M is determined by the number of orthogonal waveforms that the system can transmit simultaneously without interference. Note that in the classical MAB problem, $M = 1$. Tasks are selected over time according to a Markovian prioritization policy u (which selects the current action as a function of the current state). Let $\mathscr{U}$ denote the class of admissible Markovian policies. The prioritization policy $u \in \mathscr{U}$ is $[a^k_n] \in \{0,1\}^{N \times K}$. The TPRB problem consists of finding a prioritization policy that maximizes the total expected discounted reward over an infinite horizon (we assume K is large enough for the infinite-horizon approximation) subject to the constraint on the number of tasks that must be selected and the dynamical constraint of each of the Markov decision processes represented by a tuple $< P^0, P^1, R^0, R^1, \beta >$. The total expected discounted reward objective function can be described as

$$\max_{u \in \mathscr{U}} \quad Z = E_u \left[\sum_{k=0}^{\infty} \sum_{n=1}^{N} \beta^k R^{a^k_n}_{i^k_n} \right]. \tag{3}$$

5 THE TPRB SCHEME

In this section, we solve the TPRB problem mentioned in Section 4 using the primal-dual index heuristic algorithm based on a first-order relaxation to form the TPRB scheme which has been demonstrated to have less complexity and nearly optimal performance (Bertsimas and Niño-Mora 2000).

5.1 A First-order LP Relaxation

The first order relaxation as proposed by Whittle states that the original requirement that exactly M tasks must be selected at any ping interval is relaxed to an averaged version: the total expected discounted number of active tasks must be $M/(1-\beta)$ (Whittle 1998). According to Whittle this relaxed version must be interpreted as the problem of controlling optimally N separate Markov decision processes, subject to the binding constraint on the average number of active tasks. In this section Whittle's relaxation is reformulated using a linear programming model developed in Bertsimas and Niño-Mora (2000).

The convenient decision variables (denoted here by $x^{a_n}_{i_n}$) for a linear programming model for the Markov decision processes are defined as follows. For each $n \in \mathscr{N}$, $i_n \in \mathscr{S}_n$ and $a_n \in \{0,1\}$, let $x^{a_n}_{i_n}$ be the expected total discounted time that task n is in state i_n and action a_n is made, when the distribution of the initial state is known and given by a vector α; i.e.,

$$x^{a_n}_{i_n} = E_u \left[\sum_{k=0}^{\infty} I^{a^k_n}_{i^k_n} \beta^k \right], \tag{4}$$

where

$$I^{a^k_n}_{i^k_n} = \begin{cases} 1 & \text{if task } n \text{ is in state } i^k_n \text{ and action } a^k_n \text{ is taken at interval } k, \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

The initial states of the tracking tasks are assumed to be known from the output of the tracker, and the vector α is represented as

$$\alpha_{i_n} = \begin{cases} 1 & \text{if project } n \text{ is in state } i_n \text{ at interval } k = 0, \\ 0 & \text{otherwise.} \end{cases} \tag{6}$$

The objective function in (3) can then be translated into

$$\max \quad Z = \sum_{n\in\mathcal{N}} \sum_{i_n\in\mathcal{S}_n} \sum_{a_n\in\{0,1\}} R_{i_n}^{a_n} x_{i_n}^{a_n}, \tag{7}$$

and the value Z is interpreted as the expected total discounted cost. Applying classical results on the long-run properties of Markov decision processes, the constraints on $x_{i_n}^{a_n}$ are given by

$$\sum_{a_n\in\{0,1\}} x_{i_n}^{a_n} - \beta \sum_{i_n\in\mathcal{S}_n} \sum_{a_n\in\{0,1\}} p_{i_n j_n}^{a_n} x_{i_n}^{a_n} = \alpha_{j_n}, \text{ for } j_n \in \mathcal{S}_n, n \in \mathcal{N}, \tag{8}$$

$$x_{i_n}^{a_n} \geq 0, \text{ for } a_n \in \{0,1\}, i_n \in \mathcal{S}_n, n \in \mathcal{N}. \tag{9}$$

Now, Whittle's condition on the average number of active tasks can be written as

$$\sum_{n\in\mathcal{N}} \sum_{i_n\in\mathcal{S}_n} x_{i_n}^1 = \sum_{k=0}^{\infty} E_u \left[\sum_{n\in\mathcal{N}} \sum_{i_n^k\in\mathcal{S}_n} I_{i_n^k}^1 \right] \beta^k = \sum_{k=0}^{\infty} M\beta^k = \frac{M}{1-\beta}. \tag{10}$$

The objective function (7) and the constraints (8) – (10) form a first-order LP relaxation model, denoted by (P^1), of the TPRB problem.

5.2 Primal-Dual Index Heuristic

To find each action a_n^k once the $x_{i_n}^{a_n}$ values are obtained, we use the primal-dual index heuristic proposed in (Bertsimas and Niño-Mora 2000). The dual of the linear program (P^1) is given by

$$\begin{aligned}
(\mathrm{D}^1) \qquad & \min \quad Y = \sum_{n\in\mathcal{N}} \sum_{j_n\in\mathcal{S}_n} \alpha_{j_n} y_{j_n} + \frac{M}{1-\beta} y \\
& \text{subject to} \\
& y_{i_n} - \beta \sum_{j_n\in\mathcal{S}_n} p_{i_n j_n}^0 y_{j_n} \geq R_{i_n}^0, \text{ for } i_n \in \mathcal{S}_n, n \in \mathcal{N}, \\
& y_{i_n} - \beta \sum_{j_n\in\mathcal{S}_n} p_{i_n j_n}^1 y_{j_n} + y \geq R_{i_n}^1, \text{ for } i_n \in \mathcal{S}_n, n \in \mathcal{N}.
\end{aligned} \tag{11}$$

We denote $\{\bar{x}_{i_n}^{a_n}\}$ and $\{\bar{y}_{i_n}, \bar{y}\}$ as an optimal and dual solution pair to the primal problem (P^1) and its dual problem (D^1) respectively. We denote $\bar{\gamma}_{i_n}^{a_n}$ as the corresponding optimal reduced cost coefficients which are given by

$$\begin{aligned}
\bar{\gamma}_{i_n}^0 &= \bar{y}_{i_n} - \beta \sum_{j_n\in\mathcal{S}_N} p_{i_n j_n}^0 \bar{y}_{j_n} - R_{i_n}^0, \\
\bar{\gamma}_{i_n}^1 &= \bar{y}_{i_n} - \beta \sum_{j_n\in\mathcal{S}_N} p_{i_n j_n}^1 \bar{y}_{j_n} + \bar{y} - R_{i_n}^1,
\end{aligned} \tag{12}$$

where $\bar{\gamma}_{i_n}^0, \bar{\gamma}_{i_n}^1 \geq 0$. $\bar{\gamma}_{i_n}^0$ and $\bar{\gamma}_{i_n}^1$ can be interpreted as the rates of decrease in the objective function value of the primal problem (P^1) per unit increase in the value of variables $x_{i_n}^0$ and $x_{i_n}^1$ respectively.

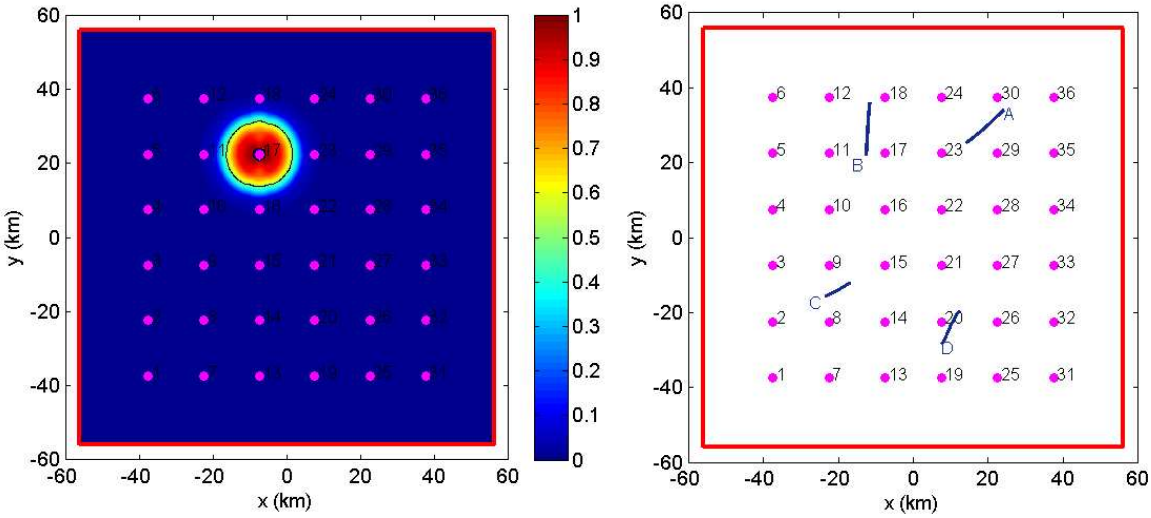
Given the current states $(i_1, \ldots, i_N)$ of the N tasks, the index for each of the task when it is in state i_n is then computed as

$$\zeta_{i_n} = \bar{\gamma}_{i_n}^1 - \bar{\gamma}_{i_n}^0. \tag{13}$$

According to the primal-dual index heuristic, the tasks that have the M smallest indices are set active. In case of ties, tasks with $\bar{x}_{i_n}^1 > 0$ are set active.

6 SIMULATION RESULTS

In this section we illustrate the performance of the TPRB policy on multi-target scenarios using active sonar tracking system simulations (Section 2). A monostatic active system with 36 nodes distributed in an operational region of approximately 120×120 km^2 in a 6×6 grid with a spacing of 15 km between neighboring nodes is used. The field layout and the probability of detection map associated with a node are shown in Figure 4a. Randomly-generated tracks are implemented to simulate tracking tasks. A 4-target tracking scenario example is illustrated in Figure 4b. The number of tracks vary for different case studies.



(a) (b)

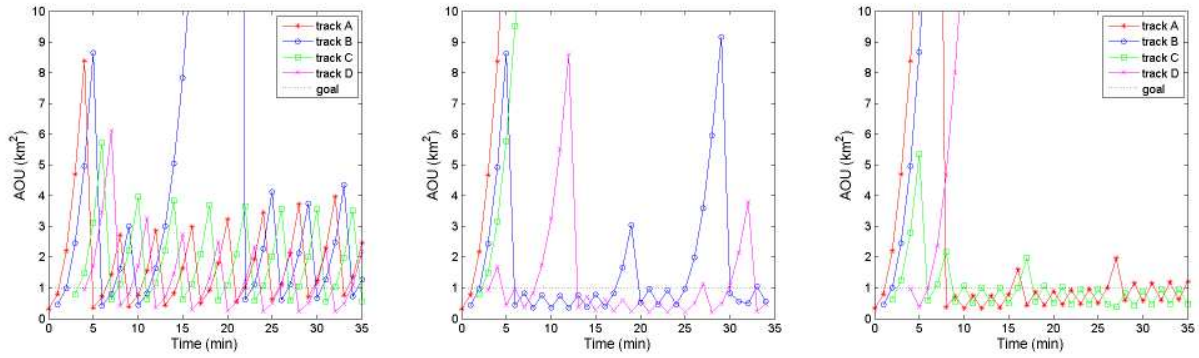
Figure 4: (a) Probability of detection map of sonar node 17. (b) A 4-target tracking scenario example.

For performance comparison we implement traditional task prioritization policies including a round robin policy and a greedy policy in addition to the TPRB policy. In this study, we set $M = 1$ assuming only one ping from one node can be transmitted at each ping interval. The round robin policy uses a pre-specified sequence for the N tracks to determine the ping decisions, i.e., at each ping time a ping is transmitted from the node that will result in a minimum AOU and a detection probability of greater than 0.5 for the selected track. The greedy policy selects a track with the largest reward, denoted by n^*, from the N tracks at each ping time, i.e., $n^* = \text{argmax}_{n \in \mathcal{N}}(R_n^1 + \sum_{i \in \mathcal{N}, i \neq n} R_i^0)$. As currently implemented, the simulation experiments were performed using MATLAB. The linear programming models were solved using LP_Solve 5.5 (LP_Solve 2015).

The process flow of the active sonar tracking system simulation can be summarized as follows: given the number of tracking tasks and their positions and velocities, pings are first generated according to a round robin policy for track initialization. Task prioritization is initialized when at least one detection from each track has been obtained. The task prioritization model (round robin, greedy, or TPRB policy) is solved to select a set of tracking tasks to perform next. Based on the selected tasks, the scheduling model provides the ping decisions using sonar performance model predictions, and pings are generated accordingly. Detections from the pings are processed by the multi-target tracker and the AOU state estimates of the tracking tasks are updated. At this point the optimization procedure iterates. The task prioritization and scheduling models are solved using the updated tracking tasks to generate new ping decisions. This process is repeated until the end of the operational scenario time window.

To gain insights into the effect of task prioritization on the tracking performance, a 4-target tracking scenario example (Figure 4b) is presented. In this example, it is desired to maintain tracks with their AOU values below 1 km^2 (i.e., $G_{\text{aou}} = 1$ km^2). Typically, the goal AOU threshold G_{aou} will be selected by the

system's operator. The AOU performance on the tracks with the round robin, greedy and TPRB policies are plotted in Figure 5. To initialize task prioritization, pings are first generated according to a round robin policy. At 5 minutes into the scenario, all the 4 tracks have been initialized and task prioritization is being performed for the duration from 5 minutes to 35 minutes. With each ping the AOU of the selected track decreases if it results in a detection and increases if it results in a miss, while the AOUs of the non-selected tracks increase. The total number of detections over the scenario duration are the same for each of the policies. However, the round robin policy cannot maintain continuous tracks below the goal AOU threshold of 1 km^2. The greedy and TPRB policies strive to maintain 2 continuous tracks below 1 km^2 threshold. In this example, the greedy policy schedules a ping for an additional track at 30 minutes yielding an inferior tracking result to the TPRB policy.



(a) (b) (c)

Figure 5: The AOU trajectories of the tracks in the 4-target scenario example with (a) the round robin policy, (b) the greedy strategy, and (c) the TPRB policy.

The tracking performance of the TPRB, greedy and round robin policies is evaluated for N-target scenarios, where $N = 3,4$, and $G_{\text{aou}} = 0.5, 0.7, 1, 3$ km^2. Setting G_{aou} to a smaller value imposes a more stringent requirement on the sonar tracking system as each tracking task demands more pings. For each scenario, N random tracks are generated. Performance evaluation of each policy for each case study is based on 10 N-target scenarios with 10 trials for each scenario (a total of 100 trials). The other simulation parameters are defined as follows: the ping interval is set to 1 minute. The number of pings at each interval is set to $M = 1$. The AOU state of each tracking task is quantized into 15 states with the corresponding AOU ranges shown in Table 1. The values associated with being in each state is set to $V_i = v$ for each state $i = 1, \ldots, 15$, where $v = 40$ for the desirable states (states with their AOU values below G_{aou}), $v = -1, -2, \ldots$ for each deteriorating state thereafter, with $v = -15$ for $i = 15$. The time discounting parameter for the TPRB policy is set to $\beta = 0.6$. Each scenario is initiated with a round robin policy, and the task prioritization module is activated when N tracks have been detected. The task prioritization duration is set to 30 minutes.

In Table 2 we report the results of our experiments for various values of the parameters. For a given set of parameters, the following quantities are computed for each of the policies: Z_{aou}: mean AOU value of the portions of the output tracks with values below G_{aou} (in km^2), n_T: mean number of successfully tracked targets with AOU values below G_{aou}, n_Γ: mean number of track fragments, d_Γ: mean duration of track fragments (in minutes), and D_Γ: mean of the maximum duration of track fragments (in minutes) in each trial. In general, the values of $Z_{\text{aou}} <= G_{\text{aou}}$ are of equivalent performance. With n_T, d_Γ, and D_Γ, the higher the values, the more desirable the system performance is. With n_Γ, the smaller values are more desirable due to less track fragmentation.

It is observed that in general the performance of the greedy and TPRB policies significantly improves the track quality of the prioritized tracks in terms of track fragmentation at the expense of neglecting tracking

tasks that cannot be fulfilled. The mean numbers of successfully tracked targets (n_T) are comparable among all policies except for the cases with a stringent goal AOU threshold ($G_{\text{aou}} = 0.5$) or an easy goal AOU threshold ($G_{\text{aou}} = 3$). In the cases with $G_{\text{aou}} = 0.5$, without any successive detection opportunities on a single track, the round robin policy struggles to bring the AOU values below the threshold. In the case with $G_{\text{aou}} = 3$ and $N = 3$, the greedy and TPRB policy is overly conservative by putting too much weights on maintaining quality tracks and their overall performance is inferior to that of the round robin policy, suggesting that the tracking system can provide performance without task prioritization. The value of task prioritization in terms of track quality parameters ($n_\Gamma, d_\Gamma, D_\Gamma$) is apparent when the requirement for the tracking tasks exceeds available ping resources. It can be observed that the greedy policy significantly improves track quality over the round robin policy except for the case with $G_{\text{aou}} = 3$ and $N = 3$. Although not by much, the greedy policy even outperforms the TPRB policy for the cases with $G_{\text{aou}} = 0.5$. In all the other cases, the TPRB policy provides at least an equivalent or better performance over the greedy policy in terms of track fragmentation metrics. Since the TPRB policy provides a more consistent performance with limited ping resources, it can be concluded that the TPRB policy is superior to that of the round robin and greedy policies.

Table 2: Numerical Experiments

Parameters	Policies	Z_{aou}	n_T	n_Γ	d_Γ	D_Γ
$N=4$, $M=1$, $G_{\text{aou}}=0.5$	TPRB	0.29	0.92	3.04	8.85	20.81
	Greedy	**0.28**	**0.95**	**2.44**	**11.31**	**21.99**
	Round robin	0.38	0.47	26.14	0.15	0.41
$N=4$, $M=1$, $G_{\text{aou}}=0.7$	TPRB	**0.32**	0.97	**3.90**	**6.70**	**20.54**
	Greedy	0.38	**0.99**	9.01	2.33	11.11
	Round robin	0.48	0.95	25.37	0.18	0.96
$N=4$, $M=1$, $G_{\text{aou}}=1$	TPRB	**0.58**	**1.60**	**4.73**	**5.35**	20.05
	Greedy	0.59	1.51	5.13	4.84	**20.53**
	Round robin	0.61	1.40	19.11	0.57	1.19
$N=4$, $M=1$, $G_{\text{aou}}=3$	TPRB	**0.82**	1.99	**3.83**	**6.83**	**20.49**
	Greedy	1.22	2.77	8.36	2.59	9.95
	Round robin	1.22	**2.81**	7.89	2.84	17.31
$N=3$, $M=1$, $G_{\text{aou}}=0.5$	TPRB	0.29	0.87	3.87	6.75	17.16
	Greedy	**0.28**	**0.89**	**3.63**	**7.27**	**17.67**
	Round robin	0.38	0.48	28.19	0.07	0.21
$N=3$, $M=1$, $G_{\text{aou}}=0.7$	TPRB	**0.34**	0.93	**4.74**	**5.33**	**17.76**
	Greedy	0.39	**0.95**	6.84	3.38	13.37
	Round robin	0.48	0.92	24.99	0.20	0.71
$N=3$, $M=1$, $G_{\text{aou}}=1$	TPRB	**0.56**	**1.42**	**5.90**	**4.09**	15.59
	Greedy	0.58	1.35	6.67	3.50	**16.31**
	Round robin	0.59	1.36	19.11	0.57	1.48
$N=3$, $M=1$, $G_{\text{aou}}=3$	TPRB	**0.87**	1.90	4.04	6.42	19.51
	Greedy	1.18	**2.51**	4.55	5.59	16.83
	Round robin	1.12	**2.51**	**2.71**	**10.06**	**27.34**

7 CONCLUSIONS

In this paper, we addressed the dynamic task prioritization problem for an active sonar tracking system. Finite-state Markov chains that capture the dynamical behavior of tracks' AOUs are derived from the sonar system simulations. The task prioritization problem is established as a restless bandit problem, which is solved by the primal-dual index heuristic algorithm based on a first-order relaxation to form the TPRB scheme. Extensive simulation results illustrate the superior performance of the proposed scheme over the conventional round robin approach in terms of track fragmentation especially when the system is overloaded with a large number of tracking tasks and/or has a stringent AOU requirement for the tracking tasks.

REFERENCES

Bar-Shalom, Y., P. K. Willett, and X. Tian. 2011. *Tracking and Data Fusion: A Handbook of Algorithms*. Storrs, CT: YBS Publishing.

Bertsimas, D., and J. Niño-Mora. 2000. "Restless Bandits, Linear Programming Relaxations and a Primal-Dual Heuristic". *Operations Research* 48:80–90.

Gittins, J. 1979. "Bandit Processes and Dynamic Allocation Indicies (with discussion)". *Journal of Royal Statistical Society* 41 (2): 148–177.

Grimmett, D., and S. Coraluppi. 2006. "Multistatic Active Sonar System Interoperability, Data Fusion, and Measures of Performance". Technical Report NURC-FR-2006-004, NATO Undersea Research Centre.

LP_Solve 2015. "LP_Solve Reference Guide Menu". Accessed Feb. 1, 2015. http://lpsolve.sourceforge.net.

Mahajan, A., and D. Teneketzis. 2008. "Multi-Armed Bandit Problems". In *Foundations and Applications of Sensor Management*, edited by A. O. H. III, D. A. Castanon, D. Cochran, and K. Kastella, 121–151. Springer-Verlag.

MSPM 2010. "Multistatic Sonar Performance Model". [software] (version 1.4). Signal Systems Corporation, Maryland.

Wakayama, C. Y., Z. B. Zabinsky, R. Plate, and J. Hoff. 2015. "Dynamic Adaptive Ping Scheduling for Monostatic Active Sonar Systems in Convergence Zone Environments". *U.S. Navy Journal of Underwater Acoustics* (accepted).

Whittle, P. 1998. "Restless Bandits: Activity Allocation in a Changing World". *Journal of Applied Probability* 25:287–298.

AUTHOR BIOGRAPHIES

CHERRY WAKAYAMA is a Research Engineer in the Maritime Systems Division at SPAWAR Systems Center Pacific. She is particularly involved in research and development concerning multistatic active sonar information fusion, target tracking and sensor management. She holds M.S. and Ph.D. degrees in Electrical Engineering from the University of Washington. Her research interests include dynamical systems, optimal control theory and applications of operations research. Her e-mail address is wakayama@spawar.navy.mil.

ZELDA B. ZABINSKY is a Professor in the Department of Industrial and Systems Engineering at the University of Washington, with adjunct appointments in the departments of Electrical Engineering, Mechanical Engineering, and Civil and Environmental Engineering. She is an IIE Fellow. Professor Zabinsky's research interests are in global optimization under uncertainty for complex systems. Her book, *Stochastic Adaptive Search in Global Optimization*, describes research on theory and practice of algorithms useful for solving problems with multimodal objective functions in high dimension. Her email address is zelda@u.washington.edu.