Self-Adaptive Resource Allocation for Event Monitoring with Uncertainty in Sensor Networks

Nan Hu and Thomas La Porta
Department of Computer Science and Engineering
The Pennsylvania State University
Email: nqh5045@psu.edu

Novella Bartolini
Computer Science Department
Sapienza University of Rome, Italy
Email: bartolini@di.uniroma1.it

***Abstract*—Event monitoring is an important application of sensor networks. Multiple parties, with different surveillance targets, can share the same network, with limited sensing resources, to monitor their events of interest simultaneously. Such a system achieves profit by allocating sensing resources to missions to collect event related information (e.g., videos, photos, electromagnetic signals). We address the problem of dynamically assigning resources to missions so as to achieve maximum profit with uncertainty in event occurrence. We consider time-varying resource demands and profits, and multiple concurrent surveillance missions. We model each mission as a sequence of monitoring attempts, each being allocated with a certain amount of resources, on a specific set of events that occurs as a Markov process. We propose a Self-Adaptive Resource Allocation algorithm (SARA) to adaptively and efficiently allocate resources according to the results of previous observations. By means of simulations we compare SARA to previous solutions and show SARA's potential in finding higher profit in both static and dynamic scenarios.**

I. Introduction

Sensor networks have a great potential to support a variety of applications, one of which is to monitor special events [1], [2], [3]. Constrained by the functionality and availability of sensing resources, it is sometimes prohibitive to monitor every event simultaneously. Thus, an efficient resource allocation solution is needed to monitor the most valuable events.

Heavy attention has been paid to this allocation problem in many novel sensor network applications. The approaches proposed in [4] and [5] attempt to guarantee a minimum degree of sensing coverage while optimizing either the number of active sensors or power consumption. In this paper, we consider surveillance missions which target a specific set of events that may occur with some probability; a certain amount of resources is required to monitor each event. If the required resources have been allocated when an event occurs, the mission will collect related information and achieve an associated profit. We assume that the availability of resources can be quantitatively measured [6], and resources can be either separated to support multiple missions, or combined together to statisfy higher resource-demanding missions.

If the events that will occur, as well as their demands on resources and profits, are known a priori, this resource allocation problem can be formulated as the general Knapsack Problem, for which both Polynomial-Time Approximation Scheme (PTAS) and Fully PTAS (FPTAS) have been proposed. In practice, however, we cannot perfectly predict which events will occur. As a consequence, the actual demands of resources and profit achieved may differ from what is predicted. In this paper, we aim at achieving the optimal profit through successful observations, when the events that will occur are uncertain.

Consider a mission, which initially requires one camera to monitor the surroundings. When an event (e.g., smoke or alarm) is observed, more resources (e.g., additional cameras or sensors) may be allocated for a better understanding of the ongoing condition. If the amount of allocated resources is insufficient, the mission may return no profit due to a failed observation attempt; instead if too many resources are allocated and exceed what is required, a portion of the resources is over allocated and therefore wasted. Our work tries to balance the risk of insufficient assignment and the drawback of over allocation.

We model a mission as a sequence of observation attempts on specific events of interest, and assume that the probability that one event follows another is known based on a preliminary statistical analysis. For each mission, the event occurrence can be modeled as a Markov process, each state of which is represented by an occurring event. Depending on whether the allocated resources are sufficient to monitor the occurring event, the observation in that state can be successful or not.

In our system, the missions may be submitted before the system starts or during the system's lifetime. Some works consider instant acceptance and rejection of newly submitted missions [7] while others maintain every mission in a list until its deadline arrives [8]. We consider the former approach too aggressive because a mission that is currently low-profit may contribute more in the future if a high-valued event occurs. In this paper, all missions are pooled and wait for resources to be allocated, while the system keeps updating the conditions of event occurrence and adjusts the resource allocation plan accordingly. We propose an algorithm SARA (i.e., Self-Adaptive Resource Allocation) to guide the dynamic allocation of resources to missions.

Our contributions include:

- For missions monitoring specific events, we introduce a mission model based on an event-driven Markov process, to evaluate the value of missions in different conditions, even when the events that will occur are uncertain;

- We develop an algorithm SARA, which takes the mission model as input, and adaptively adjusts resource allocation solutions along with the evolving conditions;
- We perform numerical simulation and compare SARA with other competitive works to prove its efficiency.

The rest of this paper is organized as follows. Section II lists related work on resource allocation in sensor networks. Section III explains our mission model and problem formulation. Section IV describes the details of our algorithm SARA. Section V shows the results of simulation. Section VI is the conclusion.

II. Related Work

In the literature, there are several works addressing resource allocation problems to facilitate a wide range of applications in sensor networks. Unlike SARA, whose objective is to search for the maximum profit, some works consider power to be the most critical resource and make effort to extend the lifetime of network while providing a certain level of functionality. Cardei *et al*. [9] and Hsin *et al*. [10] propose algorithms which provide power efficiently while maintaining complete network coverage. Kumar *et al*. [11] develop an energy-efficient solution to form an impenetrable barrier and extend the algorithm for more complicated scenarios with heterogenous sensors. Cao *et al*. [12] introduce their power-saving protocol which guarantees a bounded-delay sensing coverage. Carle *et al*. [13] and Wang *et al*. [14] study the scenarios where both surveillance coverage and network connectivity are required to be guaranteed.

Rather than providing complete sensing coverage, some works focus on the targets in which the users are interested. Gui *et al*. [15] propose a collaborative messaging scheme among sensors to track the movement of a single target, while Liu *et al*. [16] monitor all permanent targets as long as possible. In our settings, we handle multiple missions simultaneously instead of supporting a single tracking mission; we also consider that no event can be guaranteed to be observed all the time.

We also consider the uncertainty in event occurrence, which leads to uncertain demands and profits of missions before the events actually occur. Our previous work [17] proposes an algorithm for a static stochastic resource allocation problem. Mainland *et al*. [18] design a decentralized self-organized approach which lets the sensors independently decide their actions according to their previous performance. Fang *et al*. [8] study the sensor activation problem and suggest optimal scheduling based on what happened in the past. Our algorithm SARA takes a Markov process-based mission model as input, evaluates the value of missions based on known event occurrence conditions, and tunes the allocation plan accordingly.

III. Mission Model and Problem Formulation

In this paper, we assume that each surveillance mission has many targets of interest, and define an *event* to be a single or any combination of these targets, such that the set of targets occurring at anytime corresponds to an event. Therefore only one event occurs at a time. The event occurrence can be modeled as a discrete time Markov process, for which we identify the state as the event currently needing sensing resources, and the transitions between states as the occurrence of a new event. As an example, consider a mission that requires the monitoring of potential fires which may or may not be followed by explosions or by harmful smoke, or by both. In such a scenario, the occurrence of a fire may be modeled as event e_1, potentially followed by several repetitions of the same event e_1 in the following time slots. A successive explosion would bring the mission to state e_2, while instead the smoke would cause a transition to e_3. The occurrence of both smoke and explosions would require a major resource expenditure for simultaneously monitoring two targets and would be considered as a different event e_4.

We model the lifetime of a mission based on this event-driven Markov process, where a mission is considered as a sequence of attempts at monitoring the occurring events. At each time slot, the required resources and achievable profit of a mission are consequent to the state of the process (i.e., the occurring event). In the following we will interchangeably adopt the terms "state" and "event" of a mission.

Table I lists the frequently used notations.

TABLE I
Frequently Used Notations

Notations	Descriptions
$m_i, e_j^{(i)}, EOI_i$	Mission i, j-th event of m_i, set of m_i's events
$d_j^{(i)}, v_j^{(i)}$	Demand and profit of $e_j^{(i)}$
$\vec{D}_i, \vec{\mathcal{D}}$	Vector of $d_j^{(i)}$, vector of $\vec{D}_i$
$\vec{V}_i, \vec{\mathcal{V}}$	Vector of $v_j^{(i)}$, vector of $\vec{V}_i$
$\mathcal{P}_i, \mathcal{P}_i^{\Delta t}$	Transition matrix of m_i, $\mathcal{P}_i$ to the power of Δt
$p_{jk}^{(i)}(\Delta t), p_{jk}^{(i)}$	Probability that $e_k^{(i)}$ occurs exactly Δt time slot(s) after $e_j^{(i)}$; note that $p_{jk}^{(i)} = p_{jk}^{(i)}(1)$
$\vec{\Pi}_i, \pi_j^{(i)}$	Long-term probability distribution of EOI_i, long-term occurrence probability of $e_j^{(i)}$
$\theta \in [0, 1]$	Required minimum successful observation rate for activated missions
$\vec{U}_i(r, \Delta t)$	Expected profit of m_i in Δt time slot(s) with an amount r of resources, given different initial events
$\vec{R}(t), r_i(t)$	Resource allocation strategy at time t, the amount of resources allocated to m_i at time t
$G_{opt}(t_1, t_2)$	Actual best profit achievable between time t_1 and t_2

More formally, we consider α missions $\{m_1, \cdots, m_\alpha\}$. The lifetime of mission m_i evolves through $(n_i + 1)$ possible states corresponding to the elements of the set Events Of Interest $EOI_i \triangleq \{e_1^{(i)}, \cdots, e_{n_i}^{(i)}\}$ or to the null event $e_0^{(i)}$ that represents the case when nothing occurs.

We denote with $e_j^{(i)}$ the j-th event of mission m_i, which is characterized by a specific resource demand $d_j^{(i)}$ and by a profit $v_j^{(i)}$ which corresponds to the potential gain that the successful observation of event $e_j^{(i)}$ can contribute to the mission. To achieve $v_j^{(i)}$, when $e_j^{(i)}$ is occurring at least a resource of $d_j^{(i)}$ is required to be allocated to m_i; otherwise, the monitoring attempt will fail, resulting in zero profit by m_i in that time slot. Missions that are allocated with resources are called *activated*, and those that successfully observe events are called *valid*. We hereby denote with $\vec{D}_i = \{d_0^{(i)}, \cdots, d_{n_i}^{(i)}\}$ and $\vec{V}_i = \{v_0^{(i)}, \cdots, v_{n_i}^{(i)}\}$, the vectors of demands and profits of EOI_i, respectively.

We also denote with $p_{jk}^{(i)}$ the transition probability between events $e_j^{(i)}$ and $e_k^{(i)}$, that is the probability that, while being in state $e_j^{(i)}$ the event that occurs in the following time slot will be $e_k^{(i)}$. We hereby denote with $\mathcal{P}_i$ the transition matrix, whose elements are the transition probabilities between states.

Moreover, we denote with $p_{jk}^{(i)}(\Delta t)$ the probability that the transition between states $e_j^{(i)}$ and $e_k^{(i)}$ occurs in exactly Δt time slot(s). This value is the element of the j-th row and k-th column of the matrix $\mathcal{P}_i^{\Delta t}$ (i.e., $\mathcal{P}_i$ to the power of Δt).

We model the missions so that any of their events may occur before or after their null events, which makes the related Markov process irreducible and aperiodic (the proof is trivial). In other words, for any pair of events $e_j^{(i)}$ and $e_k^{(i)}$, the probability that $e_k^{(i)}$ will occur after $e_j^{(i)}$, given a long enough time, is non zero (i.e., $\exists \Delta t > 0$, s.t. $p_{jk}^{(i)}(\Delta t) > 0$, $\forall i, j, k$).

A. Problem Formulation

Our objective is to search for a resource allocation strategy to maximize profit achieved during the system's lifetime, subject to the constraint where the capacity of resources during one time slot is $\mathcal{C}$. $\vec{R}(t) = \{r_1(t), \cdots, r_\alpha(t)\}$ denotes the strategy applied at time t, indicating that a resource of $r_i(t)$ is allocated to mission m_i in the t-th time slot.

Let $\vec{\mathcal{S}}(t) = \{e_1(t), \cdots, e_\alpha(t)\}$ be the events occurring in the t-th time slot, where $e_i(t) \in EOI_i$, and $d_i(t)$ and $v_i(t)$ represent the demand and profit of $e_i(t)$, respectively. If $\vec{\mathcal{S}}(t)$ is known for $t \in [t_1, t_2)$, the problem to calculate $G_{opt}(t_1, t_2)$ (i.e., the optimal profit achieved between time t_1 and t_2) can be formulated as an Integer Linear Programming (ILP) problem:

$$
\begin{aligned}
G_{opt}(t_1, t_2) \quad &= \quad \max \sum_{t=t_1}^{t_2-1} \sum_{i=1}^{\alpha} v_i(t) * z_i(t) \qquad (1) \\
s.t. \qquad & \sum_{i=1}^{\alpha} d_i(t) * z_i(t) \leq \mathcal{C} \qquad \forall t \in [t_1, t_2)
\end{aligned}
$$

The solution $z_i(t)$'s are binary variables, and $r_i(t)$ is solved as $d_i(t) * z_i(t)$. If $z_i(t) = 1$, m_i is allocated with resource $d_i(t)$ for a profit $v_i(t)$ by successfully monitoring $e_i(t)$; otherwise, m_i is not activated and no resource will be assigned.

Because resources have to be allocated before $\vec{\mathcal{S}}(t)$ actually occurs, a prediction of $\vec{\mathcal{S}}(t)$ is needed, based on which an approximation of $G_{opt}(t_1, t_2)$ can be calculated.

B. Probability Distribution of $\vec{\mathcal{S}}(t)$

In order to estimate the events that occur in the future at time t (i.e., $\vec{\mathcal{S}}(t)$), we propose to use the latest observed event to calculate the probability distribution of $\vec{\mathcal{S}}(t)$, according to the Markov process of event occurrence. As mentioned before, for a given m_i, if $e_j^{(i)}$ is currently occurring and successfully observed, the probability that $e_k^{(i)}$ will occur after exactly Δt time slot(s) can be calculated as $p_{jk}^{(i)}(\Delta t)$.

By contrast, if no previous observation is available, as the Markov process model is both irreducible and aperiodic, it has a stationary probability distribution of states, which can be used as the estimate of future event arrivals. This distribution is independent of the initial state and can be calculated as follows:

$$
\begin{cases}
\vec{\Pi}_i \mathcal{P}_i &= \quad \mathcal{P}_i \\
\sum_{j=0}^{n_i} \pi_j^{(i)} &= \quad 1
\end{cases}
$$

where $\vec{\Pi}_i = \{\pi_0^{(i)}, \cdots, \pi_{n_i}^{(i)}\}$ indicates the long-term occurrence probability of each event of EOI_i. When the initial state $e_j^{(i)}$ is unknown, $\vec{\Pi}_i$ is taken as the occurrence probability distribution of EOI_i at any time.

In addition, inspired by the work of Fang *et al*. [8] and their concept of *Partial Observation Learning*, in our algorithm, we also introduce a step called *exploration*. We strategically select a newly-submitted or long-untouched mission to *explore*, as the occurrence state of its event is unknown or has not been updated for a while. This mission to explore will be activated with randomly assigned resources to get a chance of successful observation, by which its most recent state may be monitored, resulting in better understanding of current system status.

C. Eligible Resource Assignment

Once the estimate of $\vec{\mathcal{S}}(t)$ (i.e., the probability of events occurring in the t-th time slot) is calculated, the next step is to estimate how many resources should be allocated.

Targeted by the same mission, suppose that both events e_1 and e_2 have a 50% chance of occurring. The demands of e_1 and e_2 are 10 and 20, respectively. Allocating an amount of resources equal to the expectation of demands, which is 15 in this case, is not an efficient solution because it is insufficient to meet the requirements of e_2, while 5 units of resource will be wasted due to over allocation if e_1 occurs.

Therefore, instead of allocating resources to meet the demand expectation of each mission, we set a threshold θ on the successful observation rate to balance the tradeoff between under and over allocation. It is required that the resources will not be allocated to a mission unless they are sufficient to observe the event with a probability of at least θ. In other words, a θ-solution guarantees that an activated mission has at least θ probability to be valid.

Suppose that $\{e_0^{(i)}, \cdots, e_{n_i}^{(i)}\}$ is sorted in increasing order of demands and re-indexed. Given the initial state $e_j^{(i)}$, if an amount r of resource is allocated after exactly Δt time slot(s), the successful observation rate at that time can be calculated as the sum of $p_{jk}^{(i)}(\Delta t)$, where $k \in [0, n_i]$ and $r \geq d_k^{(i)}$. If the threshold θ is satisfied, the amount of r is *eligible*, and a mission can only be activated with eligible resources.

D. Expectation of Profit with Eligible Resources

The importance of missions is evaluated in terms of profit achieved via successful observations. Since the events that actually occur cannot be predicted perfectly, instead, we use the expectation of profit to evaluate missions.

We define $\vec{U}_i(r, \Delta t) = \{u_0^{(i)}(r, \Delta t), \cdots, u_{n_i}^{(i)}(r, \Delta t)\}$ for mission m_i to denote the expectation of profits in the next Δt time slot(s) when a resource of r is allocated, where $u_j^{(i)}(r, \Delta t)$ represents the value of expectation when the initial state is $e_j^{(i)}$. $\vec{U}_i(r, \Delta t)$ may be calculated iteratively as follows:

$$\vec{U}_i(r, \Delta t) \quad = \quad \vec{U}_i(r, \Delta t - 1) + \mathcal{P}_i^{\Delta t}\vec{V}_i - \vec{Y}_i(r, \Delta t) \qquad (2)$$

where

$$\begin{aligned} \vec{Y}_i(r, \Delta t) &= \{y_0^{(i)}(r, \Delta t), \cdots, y_{n_i}^{(i)}(r, \Delta t)\} \\ y_j^{(i)}(r, \Delta t) &= \sum_{k \in [0, n_i], r < d_k^{(i)}} p_{jk}^{(i)}(\Delta t) * v_k^{(i)} \end{aligned}$$

When $\Delta t = 0$, $\vec{U}_i(r, 0) = \vec{0}$ for all r; otherwise, $\vec{U}_i(r, \Delta t)$ is equal to the sum of $\vec{U}_i(r, \Delta t - 1)$ and the profit expected in the last time slot, (i.e., $\mathcal{P}_i^{\Delta t}\vec{V}_i - \vec{Y}_i(r, \Delta t)$), where the term $\vec{Y}_i(r, \Delta t)$ accounts for the events that are not granted enough sensing resources.

IV. Resource Allocation Algorithm

In this section, we introduce SARA, the self-adaptive resource allocation algorithm, which is supported by three sub-algorithms. Every Δt time slot(s) (i.e., *execution cycle*), based on known information, SARA repeatedly updates the probability distribution of event occurrence (ALG_EI in section IV-A), evaluates the values of missions when different levels of resources are allocated (ALG_EPC in section IV-B), and suggests a resource allocation solution for the next execution cycle (ALG_OSS in section IV-C).

A. Algorithm: EOI Inference (ALG_EI)

Algorithm ALG_EI infers the probability distribution of event occurrence. It takes three vectors $\vec{\mathcal{P}}$, $\vec{\mathcal{E}}$ and $\vec{\mathcal{L}}$ as inputs. $\vec{\mathcal{P}}$ is the vector of transition matrices of missions. $\vec{\mathcal{E}} = \{\varepsilon_i\}$ denote the last observed event ε_i by m_i, and $\vec{\mathcal{L}} = \{l_i\}$ records the time interval since ε_i was observed. If m_i has not observed anything, ε_i and l_i are set to -1.

The output $\vec{\mathcal{S}} = \{\vec{S_1}, \vec{S_2}, \cdots, \vec{S_\alpha}\}$ represents a vector of probability distributions, where $\vec{S_i} = \{s_0^{(i)}, \cdots, s_{n_i}^{(i)}\}$ denotes how likely it is that $e_j^{(i)}$ just occurred during the last time slot.

Missions are divided in two sets A and B (line 1), depending on the presence of previous observations of their events. A is the set of missions which have never observed any event before the current time, while B is the set of missions with at least one successful observation for each. For missions in A, since there is no information about the past, ALG_EI assigns $\vec{\Pi}_i$, the long-term probability distribution of event occurrence, as the value of $\vec{S_i}$ (line 4). If instead mission m_j belongs to B, its last successful observation occurred l_j time slot(s) ago, when it observed ε_j. The elements $p_{\varepsilon_j k}^{(j)}(l_j)$ of $\mathcal{P}_j^{l_j}$ represents the probability that event $e_k^{(j)}$ occurs exactly l_j time slot(s) after the initial state $e_{\varepsilon_j}^{(j)}$ (line 10). Note that l_j may be zero, when m_j observed ε_j in the last time slot. We define the corresponding $\mathcal{P}_j^0$ as an identity matrix.

We refer to Fang's concept of *Partial Observation Learning* [8] for a better understanding about the system status. The idea is to allocate resources to selected missions, regardless of how profitable they are, and *explore* their recent conditions. ξ in line 2 denotes the index of the mission to be explored. When A is not empty, ALG_EI selects one mission from A (line 6); otherwise, the mission with the longest interval since last successful observation will be marked (line 14). $\vec{S}_\xi$ is set as all -1's for other algorithms in the following sections (line 16).

Algorithm 1 EOI Inference

Input: $\vec{\mathcal{P}}$, $\vec{\mathcal{E}}$, $\vec{\mathcal{L}}$
Output: $\vec{\mathcal{S}} = \{\vec{S}_1, \cdots, \vec{S}_\alpha\}$
1: $A \leftarrow \{i | \varepsilon_i = -1\}$, $B \leftarrow \{i | \varepsilon_i \neq -1\}$
2: $\xi \leftarrow -1$
3: **for all** $i \in A$ **do**
4:     $\vec{S}_i \leftarrow \vec{\Pi}_i$
5:     **if** $\xi = -1$ **then**
6:         $\xi \leftarrow i$
7:     **end if**
8: **end for**
9: **for all** $j \in B$ and $k \in [0, n_j]$ **do**
10:     $s_k^{(j)} \leftarrow p_{\varepsilon_j k}^{(j)}(l_j)$
11: **end for**
12: **if** $\xi = -1$ **then**
13:     $l_{j^*} = \arg\max_j \{l_j | j \in B\}$
14:     $\xi \leftarrow j^*$
15: **end if**
16: $\vec{S}_\xi \leftarrow \{-1, \cdots, -1\}$

B. Algorithm: Expectation of Profit Calculation (ALG_EPC)

The profit expectation of a mission may be affected by two factors: the events that may occur and the amount of allocated resources. ALG_EI infers the probability distribution of the most recent event occurrence, and ALG_EPC evaluates how important each mission is when being allocated with a certain level of resources.

In addition to $\vec{\mathcal{S}}$ (i.e., output of ALG_EI) and $\vec{\mathcal{P}}$, ALG_EPC takes other four arguments: $\vec{\mathcal{D}}$, $\vec{\mathcal{V}}$, θ and Δt. The first two are the vectors of demands and profits of all missions, while θ is the threshold introduced in section III-C. Δt is the length of execution cycle, based on which ALG_EPC calculates the expectation of profit.

The output $\vec{\mathcal{W}}^{\mathcal{S}}(\Delta t)$ consists of $\vec{W}_i^{\mathcal{S}}(\Delta t)$'s. Given distribution $\vec{\mathcal{S}}$ of current states, $w_i^{\mathcal{S}}(d_k^{(i)}, \Delta t) \in \vec{W}_i^{\mathcal{S}}(\Delta t)$ represents the expected profit in the next Δt time slot(s) if a resource of $d_k^{(i)}$ is allocated.

ALG_EPC works in two parts. First, for every i, r and t, $\vec{U}_i(r,t)$ is calculated (line 12). As mentioned in section III-D, given initial state $e_j^{(i)}$, $u_j^{(i)}(r,t) \in \vec{U}_i(r,t)$ denotes m_i's expected profit in the next t time slot(s) with a resource of r allocated. Second, each element of $W_i^{\mathcal{S}}(r,\Delta t)$, except for the one that corresponds to the mission to explore, is updated by $\sum_{j=0}^{n_i} s_j^{(i)} * u_j^{(i)}(r,\Delta t)$, which is the sum of profit expectations with different initial states, where $s_j^{(i)} \in \vec{S}_i$ is the probability that $e_j^{(i)}$ is the initial event (line 16). For the mission to be explored, where $s_j^{(\xi)} = -1$, ALG_EPC keeps every $w_\xi^{\mathcal{S}}(r,\Delta t)$ as 0 (line 15). This mission will be activated with randomly assigned resources, no matter how much profit is expected.

Algorithm 2 Expectation of Profit Calculation

Input: $\vec{\mathcal{S}}$, $\vec{\mathcal{P}}$, $\vec{\mathcal{D}}$, $\vec{\mathcal{V}}$, θ, Δt
Output: $\vec{\mathcal{W}}^{\mathcal{S}}(\Delta t) = \{\vec{W}_1^{\mathcal{S}}(\Delta t), \cdots, \vec{W}_\alpha^{\mathcal{S}}(\Delta t)\}$
1: $\vec{\mathcal{W}}^{\mathcal{S}}(\Delta t) \leftarrow \vec{0}$
2: $\vec{U}_i(r,t) \leftarrow \vec{0}$, where $i \in [1,\alpha], r \in \vec{D}_i, t \in [0,\Delta t]$
3: **for** $i = 1, \cdots, \alpha$ **do**
4:   **for** $t = 1, \cdots, \Delta t$ **do**
5:     **for** $j = 0, \cdots, n_i$ **do**
6:       $p \leftarrow 0$, $v \leftarrow 0$
7:       **for** $k = 0, \cdots, n_i$ **do**
8:         $p \leftarrow p + p_{jk}^{(i)}(t)$
9:         $v \leftarrow v + p_{jk}^{(i)}(t) * v_k^{(i)}$
10:         **if** $p \geq \theta$ **then**
11:           **if** $t = 1$ **or** $u_j^{(i)}(d_k^{(i)}, t-1) > 0$ **then**
12:             $u_j^{(i)}(d_k^{(i)}, t) \leftarrow u_j^{(i)}(d_k^{(i)}, t-1) + v$
13:           **end if**
14:         **end if**
15:         **if** $t = \Delta t$ **and** $s_j^{(i)} \geq 0$ **then**
16:           $w_i^{\mathcal{S}}(d_k^{(i)}, \Delta t)$ += $s_j^{(i)} * u_j^{(i)}(d_k^{(i)}, \Delta t)$
17:         **end if**
18:       **end for**
19:     **end for**
20:   **end for**
21: **end for**

Given mission m_i and the t-th time slot after the occurrence of the initial state $e_j^{(i)}$, two auxiliary variables p and v are introduced (line 6). While k is iterated from 0 to n_i (line 7), which indicates the increment of allocated resources from $d_0^{(i)}$ to $d_{n_i}^{(i)}$, p represents the successful observation rate, and v represents the profit expectation. Each increment enables a new event $e_k^{(i)}$ to be observable, and the chance to observe successfully is increased by $p_{jk}^{(i)}(t)$ (line 8), which is the occurrence probability of $e_k^{(i)}$ given $e_j^{(i)}$ and t. Similarly, each time v (i.e., accumulated profit expectation) is increased by $p_{jk}^{(i)}(t) * v_k^{(i)}$ (line 9), which is the product of $e_k^{(i)}$'s occurrence probability and profit.

Threshold θ sets a minimum observation rate for all activated missions. When $t = 1$, $u_j^{(i)}(r,t)$ is not assigned as v until $p \geq \theta$, to guarantee that v is achievable with an eligible allocation to m_i. When $t > 1$, the value of r should also satisfy $u_j^{(i)}(r, t-1) > 0$ (line 11); otherwise, θ is not guaranteed for at least one time slot during the given execution cycle.

C. Algorithm: One-Step Scheduling (ALG_OSS)

Taking $\vec{\mathcal{W}}^{\mathcal{S}}(\Delta t)$ from ALG_EPC, $\vec{\mathcal{D}}$, the total available sensing resources $\mathcal{C}$, and current time t as inputs, ALG_OSS applies dynamic programming to calculate the optimal solution, which is the resource allocation that maximizes the expectation of profit in one execution cycle of Δt time slot(s).

When the current solution needs to be tuned at time t, ALG_OSS is executed, and its output consists of $G_{oss}(t, t+\Delta t)$ and $\vec{R}(t) = \{r_i(t)\}$. $G_{oss}(t, t+\Delta t)$ is an approximation of $G_{opt}(t, t+\Delta t)$ in problem (1), and, as described in section III-A, $r_i(t)$ denotes the resources to be allocated to m_i.

Algorithm 3 One-Step Scheduling

Input: $\vec{\mathcal{W}}^{\mathcal{S}}(\Delta t)$, $\vec{\mathcal{D}}$, $\mathcal{C}$, t
Output: $G_{oss}(t, t+\Delta t)$, $\vec{R}(t) = \{r_1(t), \cdots, r_\alpha(t)\}$
1: $\vec{F} \leftarrow \vec{0}$, $\vec{H} \leftarrow \vec{0}$
2: **for all** $i = 1, \cdots, \alpha$ **do**
3:   **for all** $r = 1, \cdots, \mathcal{C}$ **do**
4:     $f_{ir} \leftarrow f_{(i-1)r}$
5:     **for all** $j = 0, \cdots, n_i$ **do**
6:       **if** $r \geq d_j^{(i)}$ **and** $w_i^{\mathcal{S}}(d_j^{(i)}, \Delta t) > 0$ **then**
7:         $x \leftarrow f_{(i-1)(r-d_j^{(i)})} + w_i^{\mathcal{S}}(d_j^{(i)}, \Delta t)$
8:         **if** $x > f_{ir}$ **then**
9:           $f_{ir} \leftarrow x$, $h_{ir} \leftarrow d_j^{(i)}$
10:         **end if**
11:       **end if**
12:     **end for**
13:   **end for**
14: **end for**
15: $G_{oss}(t, t+\Delta t) \leftarrow f_{\alpha\mathcal{C}}$, $r \leftarrow \mathcal{C}$
16: **for all** $i = \alpha, \cdots, 1$ **do**
17:   $r_i(t) \leftarrow h_{ir}$
18:   $r \leftarrow r - r_i(t)$
19: **end for**

$\vec{F}_i$ and $\vec{H}_i$ are introduced for dynamic programming, where $f_{ir} \in \vec{F}_i$ denotes the optimal result when the following conditions are met: 1) only the missions in $\{m_j | j \in [1,i]\}$ can be activated; 2) the total allocated resources are no more than r. Each f_{ir} is associated with an $h_{ir} \in \vec{H}_i$, which denotes the amount of resources allocated to m_i to achieve f_{ir}.

When either i or r is equal to 0, there is no way to achieve any profit (i.e., f_{0r} and f_{i0} are 0); in other cases, the way of calculating f_{ir} is either not activating m_i so that f_{ir} is assigned with the same value as $f_{(i-1)r}$ (line 4), or activating m_i with a resource of $d_j^{(i)} \in \vec{D}_i$ to get $w_i^{\mathcal{S}}(d_j^{(i)}, \Delta t) \in \vec{W}_i^{\mathcal{S}}(\Delta t)$ as profit. In the second case, ALG_OSS needs to consider all $(n_i + 1)$ allocation options of $\vec{D}_i$ to find the optimal f_{ir}. Each time the best f_{ir} is found, h_{ir} is set as the corresponding $d_j^{(i)}$. Therefore, ALG_OSS solves the optimal f_{ir} in $O(\alpha\mathcal{C}n_{max})$ time, where n_{max} is the largest value among all n_i's.

Lines 2-14 calculate f_{ir} and h_{ir} by dynamic programming, and $G_{oss}(t, t+\Delta t)$ is equal to $f_{\alpha\mathcal{C}}$ (line 15). For the mission

m_ξ to be explored, the corresponding $\vec{W}_\xi^{\mathcal{S}}(\Delta t)$ is kept as zero by ALG_EPC, therefore m_ξ will not be allocated with any resource by ALG_OSS due to the constraint in line 6.

For any given optimal f_{ir}, h_{ir} indicates how many resources should be allocated to m_i. Iterating from the initial coordinate (i, r) = (α, $\mathcal{C}$), each time we can find the best option h_{ir} for m_i (line 17). After a resource of $r_i(t) = h_{ir}$ is allocated, i and r are decreased by 1 and $r_i(t)$, respectively (line 18), to reach the next coordinate ($i-1$, $r-r_i(t)$), until $\vec{R}(t)$ is obtained.

Replacing t and $(t+\Delta t)$ by t_1 and t_2, respectively, the output $G_{oss}(t_1,t_2)$ is an approximation to the actual optimal result $G_{opt}(t_1,t_2)$, which is calculated before the events actually occur. We hereby define p_{min} as the minimum value in any transition matrix $\mathcal{P}_i$. The following theorem characterizes the difference between $G_{oss}(t_1,t_2)$ and $G_{opt}(t_1,t_2)$:

Theorem 1. *In a special case where $\theta = 0$ and $p_{min} > 0$, $G_{oss}(t_1,t_2)$ and $G_{opt}(t_1,t_2)$ satisfy:*

$$\begin{aligned} G_{opt}(t_1,t_2) &\leq \mathcal{B}(t_2-t_1) * G_{oss}(t_1,t_2) \\ \textit{where } \mathcal{B}(t_2-t_1) &= \frac{(t_2-t_1)*(1/p_{min}-1)}{1-(p_{min})^{(t_2-t_1)}} \end{aligned}$$

Proof. Define G_{max} as the maximum profit achieved by any combination of events during (t_2-t_1) time slot(s), no matter if it can occur or not. Therefore $G_{opt}(t_1,t_2) \leq G_{max}$. In addition, G_{max} and $G_{oss}(t_1,t_2)$ can be solved as ILP:

$$G_{max} = \max \sum_{t=t_1}^{t_2-1} \sum_{i=1}^{\alpha} \sum_{j=0}^{n_i} v_j^{(i)} * z_j^{(i)} \quad (3)$$

$$G_{oss}(t_1,t_2) = \max \sum_{i=1}^{\alpha} \sum_{j=0}^{n_i} w_i^{\mathcal{S}}(d_j^{(i)}, t_2-t_1) * z_j^{(i)} \quad (4)$$

Both (3) and (4) yield to the same constraints:

$$\sum_{i=1}^{\alpha} \sum_{j=0}^{n_i} d_j^{(i)} * z_j^{(i)} \leq \mathcal{C} \quad (5)$$

$$\sum_{j=0}^{n_i} z_j^{(i)} \leq 1 \qquad \forall i \in [1,\alpha] \quad (6)$$

According to ALG_EPC, when $\theta = 0$,

$$w_i^{\mathcal{S}}(d_j^{(i)}, t_2-t_1) = \sum_{k=0}^{n_i} s_k^{(i)} * u_k^{(i)}(d_j^{(i)}, t_2-t_1) \quad (7)$$

where

$$\begin{aligned} u_k^{(i)}(d_j^{(i)}, t_2-t_1) &= \sum_{t=t_1}^{t_2-1} \sum_{q=1}^{j} p_{kq}^{(i)}(t-t_1+1) * v_q^{(i)} \\ &\geq \sum_{t=t_1}^{t_2-1} p_{kj}^{(i)}(t-t_1+1) * v_j^{(i)} \\ &\geq \sum_{t=1}^{t_2-t_1} (p_{min})^t * v_j^{(i)} \quad (8) \end{aligned}$$

Therefore, from (7) and (8), we get

$$\begin{aligned} w_i^{\mathcal{S}}(d_j^{(i)}, t_2-t_1) &\geq \sum_{k=0}^{n_i} s_k^{(i)} \sum_{t=1}^{t_2-t_1} (p_{min})^t * v_j^{(i)} \\ &= v_j^{(i)}(t_2-t_1)/\mathcal{B}(t_2-t_1) \quad (9) \end{aligned}$$

Note that (3) and (4) share the same constraints, which means that if we apply the solution $\vec{Z} = \{z_j^{(i)}\}$ of problem (3) to problem (4), the constraints (5) and (6) are still satisfied. Suppose that the result of applying $\vec{Z}$ to problem (4) is G', then $G' \leq G_{oss}(t_1,t_2)$ because $G_{oss}(t_1,t_2)$ is the optimal result of problem (4). Therefore,

$$\begin{aligned} G_{max} &= \max \sum_{t=t_1}^{t_2-1} \sum_{i=1}^{\alpha} \sum_{j=0}^{n_i} v_j^{(i)} * z_j^{(i)} \\ &\leq \max \sum_{i=1}^{\alpha} \sum_{j=0}^{n_i} \mathcal{B}(t_2-t_1) * w_i^{\mathcal{S}}(d_j^{(i)}, t_2-t_1) * z_j^{(i)} \\ &= \mathcal{B}(t_2-t_1) * G' \\ &\leq \mathcal{B}(t_2-t_1) * G_{oss}(t_1,t_2) \end{aligned}$$

As a result, we have proved the following relation:

$$G_{opt}(t_1,t_2) \leq G_{max} \leq \mathcal{B}(t_2-t_1) * G_{oss}(t_1,t_2)$$

□

D. Algorithm: Self-Adaptive Resource Allocation (SARA)

Supported by the algorithms shown above, SARA allocates resources in an online environment, where the system lifetime is divided into multiple execution cycles of Δt time slot(s).

Taking mission information ($\vec{\mathcal{P}}$, $\vec{\mathcal{D}}$ and $\vec{\mathcal{V}}$), resource capacity $\mathcal{C}$, threshold θ, system lifetime T and the length of execution cycle Δt as inputs, SARA repeatedly calculates $\vec{R}(t)$.

At the beginning of each execution cycle, the probability distribution $\vec{\mathcal{S}}$ of the most recent event occurrence is calculated by ALG_EI (line 5), and the profit expectation $\vec{\mathcal{W}}^{\mathcal{S}}(\Delta t)$ is solved by ALG_EPC (line 6). As describe in section IV-A, for the mission m_ξ to be explored, the corresponding $\vec{S}_\xi$ is set to all -1's by ALG_EI. An event $e_j^{(\xi)}$ of m_ξ is randomly selected and its demand $d_j^{(\xi)}$ is allocated to m_ξ (line 9 and line 11). ALG_OSS is executed without considering the resources that are already allocated for exploration (line 10), and returns the resource allocation solution for the other missions.

At the end of each time slot, after running missions based on $\vec{R}(t)$, if a mission m_i successfully observes an event, $\varepsilon_i \in \vec{\mathcal{E}}$ is updated to show the most recent known state (line 20), while $l_i \in \vec{\mathcal{L}}$ changes to 0 as it just occurred (line 21). For those missions that fail to see anything, their values in $\vec{\mathcal{E}}$ are unchanged, but those in $\vec{\mathcal{L}}$ are increased by 1, unless the original value is -1, which means that this mission has not observed any event yet (line 23).

Algorithm 4 Self-Adaptive Resource Allocation

Input: $\vec{\mathcal{P}}$, $\vec{\mathcal{D}}$, $\vec{\mathcal{V}}$, $\mathcal{C}$, θ, T, Δt

$\vec{\mathcal{E}} \leftarrow \{-1, \cdots, -1\}$
$\vec{\mathcal{L}} \leftarrow \{-1, \cdots, -1\}$
for all $t = 0, \cdots, T$ **do**
  if $t\%\Delta t = 0$ **then**
    $\vec{\mathcal{S}} \leftarrow$ ALG_EI($\vec{\mathcal{P}}, \vec{\mathcal{E}}, \vec{\mathcal{L}}$)
    $\vec{\mathcal{W}}^{\mathcal{S}}(\Delta t) \leftarrow$ ALG_EPC($\vec{\mathcal{S}}, \vec{\mathcal{P}}, \vec{\mathcal{D}}, \vec{\mathcal{V}}, \theta, \Delta t$)
    for all $i = 1, \cdots, \alpha$ **do**
      if $\vec{S}_i = \{-1, \cdots, -1\}$ **then**
        $d_j^{(i)} \in D_i$ is randomly selected
        $\vec{R}(t) \leftarrow$ ALG_OSS($\vec{\mathcal{W}}^{\mathcal{S}}(\Delta t), \vec{\mathcal{D}}, \mathcal{C} - d_j^{(i)}, t$)
        $r_i(t) \leftarrow d_j^{(i)}$
      end if
    end for
  else
    $\vec{R}(t) \leftarrow \vec{R}(t-1)$
  end if
  Allocate resource based on $\vec{R}(t)$ and run missions
  for all $i = 1, \cdots, \alpha$ **do**
    if m_i successfully observes $e_j^{(i)}$ **then**
      $\varepsilon_i \leftarrow j$
      $l_i \leftarrow 0$
    else if $l_i > 0$ **then**
      $l_i \leftarrow l_i + 1$
    end if
  end for
end for

V. Numerical Results

In this section, we test SARA's performance in different settings, and compare it with the Activation Strategy Algorithm (ASA) developed by Fang *et al.* [8] and a variant of SARA called Stationary solution. Those algorithms all aim to allocate limited sensing resources among surveillance missions to monitor the most profitable events, when the events that potentially occur cannot be predicted precisely.

ASA constructs *unbiased esimators* to evaluate expectation of profit for missions. It ranks every mission by weighing total profit that has been achieved in the past, and selects the mission set that has accumulated more profit than any other set, assuming that it will still be more profitable in the future. In our simulation, we use (i, j) to represent the case when a resource of $d_j^{(i)}$ is allocated to m_i. When m_i observes an event with $d_j^{(i)}$, the profit estimator of (i, j) will be updated. ASA selects the most valuable (i, j), in addition to performing exploration. Like ASA, SARA also adaptively tunes its solution according to the observed events, but does not evaluate missions by their achieved profits. Instead, SARA updates the evolving probability distribution of event occurrence, by which it calculates the expectation of profit in the future.

In addition, we design a variant of SARA, assuming that the distribution Π_i can be used to represent EOI$_i$'s occurrence probability over a long enough time. The output of ALG_EI is set as $\vec{\mathcal{S}}^* = \{\vec{S}_1^*, \cdots, \vec{S}_\alpha^*\}$, where $\vec{S}_i^* = \{\Pi_i, \cdots, \Pi_i\}$, based on which ALG_EPC calculates a fixed value of $\vec{\mathcal{W}}^{\mathcal{S}^*}(\Delta t)$, resulting in a fixed resource allocation for all time slots. Therefore, this solution is called Stationary.

For SARA, we compare three different values of θ = 0, 0.5 and 1. For $\theta = 0$, SARA does not guarantee any event will be observed when allocating resources; for $\theta = 1$, SARA enables those missions allocated with resources to observe any event that actually occurs; for $\theta = 0.5$, each activated mission has at least 50% chance to observe the occurring event.

Each simulation result shown in the remaining part of this section is averaged over 10 test cases.

A. Simulation Setup

For each test case, 10 missions are created, each targeting 20 events of interest. The demands of events are randomly drawn from [1, 25], and the capacity of available resources is 100. The profits follow a bimodal distribution consisting of two Gaussian distributions $\mathcal{N}$(25, 100) and $\mathcal{N}$(75, 100), by which the events are divided into two sets of high and low values. For the null event of m_i that represents the case when no event occurs, its profit is set as 0 and demand is equal to the highest demand among EOI_i, because the case that nothing occurs can be verified only when the highest potential demand has been satisfied but still nothing is observed.

Mission m_i is associated with a matrix $\mathcal{P}_i$, which is later transformed into a Markov transition matrix. Initially, the element $p_{jk}^{(i)}$ of $\mathcal{P}_i$ is generated differently according to two models: the Dense and the Sparse Model. In the Dense Model, each $p_{jk}^{(i)}$ has a 20% chance to be zero, while this chance in the Sparse Model is set as 80%; otherwise, $p_{jk}^{(i)}$ is randomly drawn from [1, 100]. In other words, in the Dense Model when an event is occurring there are more possible subsequent events. In the Sparse Model, the occurrence of events, more or less, has some specific sequence. In addition, to make sure that $\mathcal{P}_i$ is irreducible, $p_{jk}^{(i)}$ is always forced to be greater than zero if either j or k is equal to 0, which means that the null event can either occur before or after any event. After initialization, every $p_{jk}^{(i)}$ is divided by the sum of j-th row in $\mathcal{P}_i$ to represent a valid probability indicating how likely $e_k^{(i)}$ occurs after $e_j^{(i)}$.

At the end of each time slot, all information of the occurring events is collected, based on which the actual optimal profit achievable in that time slot can be solved as a Knapsack Problem. Note that this actual optimal profit is calculated with all necessary knowledge that cannot be predict precisely, thus no algorithm can achieve better performance. We will compare the results of SARA, ASA and the Stationary solution with this actual optimal.

B. Static Cases with Fixed Mission Set

In this case, missions are submitted before the system starts and never removed until the end. Each mission can be activated or deactivated at the beginning of any execution cycle.

Setting θ as 0, Fig. 1 shows the average ratio between SARA's results and the actual optimal in different models, given different lengths of Δt of execution cycle. As shown in the figure, the shorter Δt is (i.e., higher execution frequency of SARA) the better is the performance. The performance downgrades faster when increasing Δt from 1 to 5, and finally suffers by about 6% in the Dense Model (i.e., from 55% to 52%) and 21% in the Sparse Model (i.e., from 64% to 53%), indicating that the prediction over a long time horizon is more unreliable. Since $\Delta t = 1$ gives the best performance, we always apply this length to the execution cycle.

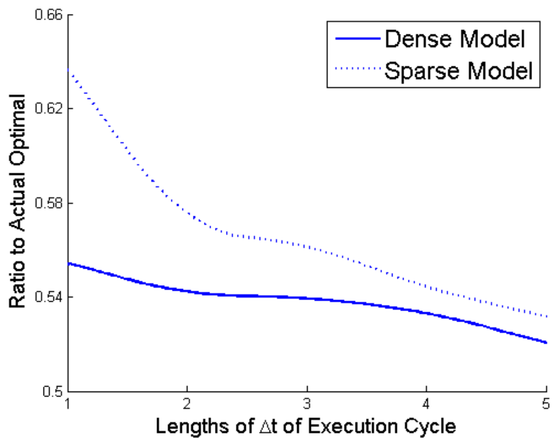

Fig. 1. SARA v.s. Actual Optimal with Variable Δt

Fig. 2 compares the performance of SARA when θ is variable. 0.5-SARA and 1-SARA are more likely to over allocate resources because a minimum observation rate needs to be guaranteed. The figure shows that SARA can perform better in the Sparse Model than in the Dense, where the event occurrence is easier to predict. 0-SARA returns better results than 1-SARA (i.e., 17% better in the Dense Model and 31% in the Sparse), but only slightly outperforms 0.5-SARA.

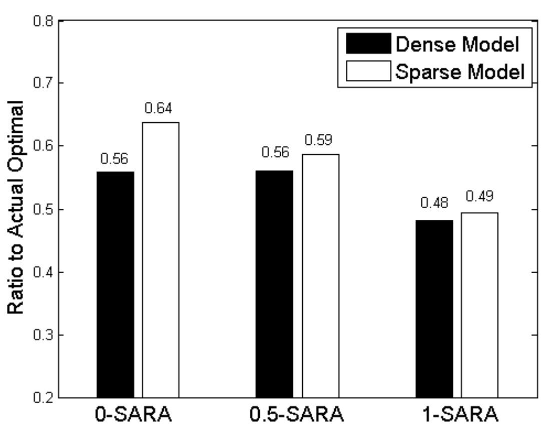

Fig. 2. SARA v.s. Actual Optimal with Variable θ

We also compare the number of valid (i.e., missions that successfully observe events) and activated missions (i.e., missions that are allocated with resources) among θ-SARAs in Fig. 3, where the white and black bars represent corresponding number, respectively, and the ratio between the numbers is marked over the set of bars. The figure shows that both 0.5-SARA and 1-SARA satisfy the required observation rate, although 1-SARA's rate is slightly lower than 100% because of exploration. 0-SARA results in a similar number of valid missions as 0.5-SARA achieves, where the former activates the most number of missions for the best profit and the latter, with a higher threshold $\theta = 0.5$, concentrates resources on fewer missions to achieve a better observation rate. 1-SARA is shown to be too conservative as its numbers of both valid and activated missions are the lowest, which results in the poorest result shown in Fig. 2.

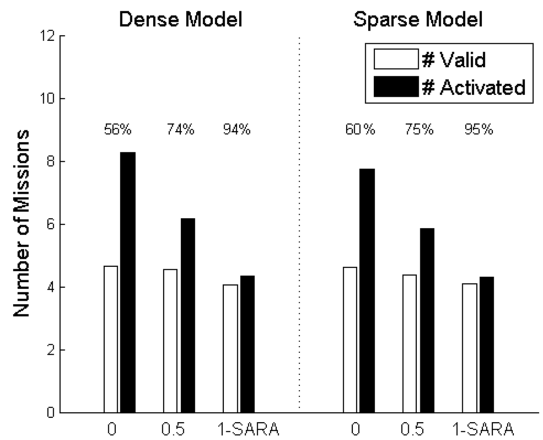

Fig. 3. Number of Valid and Activated Missions Achieved by SARAs

Fig. 4 compares 0-SARA, ASA and the Stationary solution. 0-SARA performs the best in both Dense and Sparse models.

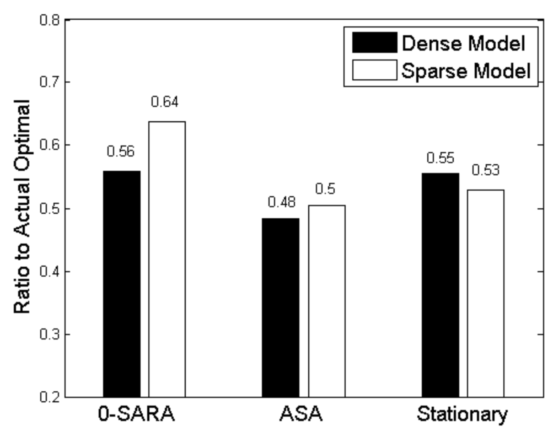

Fig. 4. 0-SARA, ASA and Stationary v.s. Actual Optimal

The difference between 0-SARA and the Stationary solution is trivial in the Dense Model, where most pairs of events can occur successively. Even if SARA observes what is currently occurring, it is hard to precisely predict what will occur in the future. As a result, over a long period of time, the results of SARA and Stationary solution may be close, but both are better than ASA by about 17% (i.e., 56% compared to 48%).

In the Sparse Model, instead, because the pairs of events that can occur one after the other are limited, based on the results of observations, SARA can modify previous allocation strategies when a mission becomes more or less important. This provides

SARA an advantage over ASA and the Stationary solutions in each time slot. SARA achieves a performance more than 28% better than ASA (i.e., averaging 64% of the actual optimal compared to 50%) and 21% better than the Stationary solution (i.e., averaging 64% compared to 53%).

C. Dynamic Cases with Changing Mission Set

In this case, 10 missions are initially created, and at the beginning of each time slot, a new mission may be submitted. The lifetime of each mission is randomly drawn from 1 to 20 time slots. When the deadline of a mission arrives, it is terminated and no longer waits to be activated.

Since ASA and the Stationary solution are designed for static scenarios and cannot be applied to this setting, in this section we only study SARAs with variable θ. Fig 5 shows that, on average, SARA still works better in the Sparse Model. In both models, 0-SARA performs better than the other two, and the difference is similar to that in the static simulation.

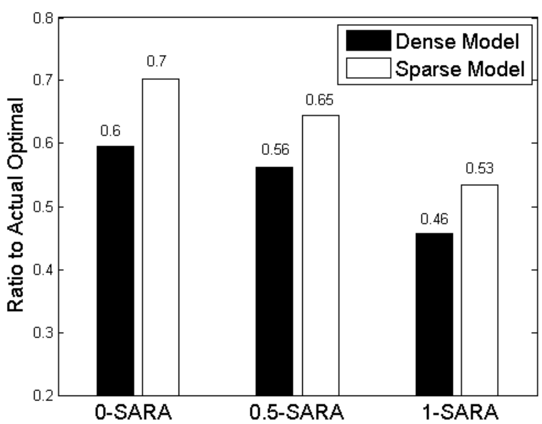

Fig. 5. Performance of SARAs in Online Environment

VI. Conclusion

We model a surveillance mission as a sequence of observation attempts on events of interest, whose occurrence is modeled as a Markov process. We develop a Self-Adaptive Resource Allocation algorithm (SARA) to maximize the expectation of profit by efficiently allocating limited sensing resources, which is also capable of tuning the allocation strategy based on the evolving conditions. Although the occurrence states of events cannot be predicted precisely, based on the known information collected from successful observations, SARA calculates an approximation result. For a special case, we prove that the difference between SARA and the actual optimal is bounded. Simulation results show that the performance of SARA is competitive compared to other algorithms in this scenario of event monitoring with uncertainty.

Acknowledgment

Research was sponsored by the U.S. Army Research Laboratory and the U.K. Ministry of Defence and was accomplished under Agreement Number W911NF-06-3-0001. The views and conclusions contained in this document are those of the author(s) and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Army Research Laboratory, the U.S. Government, the U.K. Ministry of Defence or the U.K. Government. The U.S. and U.K. Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

References

[1] E. Onur, C. Ersoy, H. Delic, and L. Akarun, "Surveillance wireless sensor networks: Deployment quality analysis," *Network, IEEE*, vol. 21, no. 6, pp. 48–53, November 2007.

[2] A. Cerpa, J. Elson, D. Estrin, L. Girod, M. Hamilton, and J. Zhao, "Habitat monitoring: Application driver for wireless communications technology," *SIGCOMM Comput. Commun. Rev.*, vol. 31, no. 2 supplement, pp. 20–41, Apr. 2001.

[3] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson, "Wireless sensor networks for habitat monitoring," in *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications*, ser. WSNA '02, New York, NY, USA, 2002, pp. 88–97.

[4] K.-P. Shih, Y.-D. Chen, C.-W. Chiang, and B.-J. Liu, "A distributed active sensor selection scheme for wireless sensor networks," in *Computers and Communications, 2006. ISCC '06. Proceedings. 11th IEEE Symposium on*, June 2006, pp. 923–928.

[5] X. Bai, Z. Yun, D. Xuan, B. Chen, and W. Zhao, "Optimal multiple-coverage of sensor networks," in *INFOCOM, 2011 Proceedings IEEE*, April 2011, pp. 2498–2506.

[6] D. Pizzocaro, A. Preece, F. Chen, T. Porta, and A. Bar-Noy, "A distributed architecture for heterogeneous multi sensor-task allocation," in *Distributed Computing in Sensor Systems and Workshops (DCOSS), 2011 International Conference on*, June 2011, pp. 1–8.

[7] T. İlhan, S. M. R. Iravani, and M. S. Daskin, "Technical note—the adaptive knapsack problem with stochastic rewards," *Oper. Res.*, vol. 59, no. 1, pp. 242–248, Jan. 2011.

[8] X. Fang, D. Yang, and G. Xue, "Strategizing surveillance for resource-constrained event monitoring," in *INFOCOM, 2012 Proceedings IEEE*, March 2012, pp. 244–252.

[9] M. Cardei, M. Thai, Y. Li, and W. Wu, "Energy-efficient target coverage in wireless sensor networks," in *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, March 2005.

[10] C.-f. Hsin and M. Liu, "Network coverage using low duty-cycled sensors: Random & coordinated sleep algorithms," in *Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks*, ser. IPSN '04, New York, NY, USA, 2004, pp. 433–442.

[11] S. Kumar, T. Lai, M. Posner, and P. Sinha, "Maximizing the lifetime of a barrier of wireless sensors," *Mobile Computing, IEEE Transactions on*, vol. 9, no. 8, pp. 1161–1172, Aug 2010.

[12] Q. Cao, T. Abdelzaher, T. He, and J. Stankovic, "Towards optimal sleep scheduling in sensor networks for rare-event detection," in *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks*, ser. IPSN '05, Piscataway, NJ, USA, 2005.

[13] J. Carle and D. Simplot-Ryl, "Energy-efficient area monitoring for sensor networks," *Computer*, vol. 37, no. 2, pp. 40–46, Feb 2004.

[14] X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, and C. Gill, "Integrated coverage and connectivity configuration in wireless sensor networks," in *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems*, ser. SenSys '03, New York, NY, USA, 2003, pp. 28–39.

[15] C. Gui and P. Mohapatra, "Power conservation and quality of surveillance in target tracking sensor networks," in *Proceedings of the 10th Annual International Conference on Mobile Computing and Networking*, ser. MobiCom '04, New York, NY, USA, 2004, pp. 129–143.

[16] H. Liu, P. Wan, C.-W. Yi, X. Jia, S. Makki, and N. Pissinou, "Maximal lifetime scheduling in sensor surveillance networks," in *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, March 2005, pp. 2482–2491.

[17] N. Hu, D. Pizzocaro, M. Johnson, T. Laporta, and A. Preece, "Resource allocation with non-deterministic demands and profits," in *Mobile Ad-Hoc and Sensor Systems (MASS), 2013 IEEE 10th International Conference on*, Oct 2013, pp. 145–153.

[18] G. Mainland, D. C. Parkes, and M. Welsh, "Decentralized, adaptive resource allocation for sensor networks," in *Proceedings of the 2nd Conference on Symposium on Networked Systems Design & Implementation - Volume 2*, ser. NSDI'05, Berkeley, CA, USA, 2005, pp. 315–328.