Tractable Planning for Coordinated Story Capture: Sequential Stochastic Decoupling

Diptanil Chaudhuri[1], Hazhar Rahmani[2], Dylan A. Shell[1], and Jason M. O'Kane[2]

[1] Dept. of Computer Science and Engineering, Texas A&M University, , College Station, TX, USA.
[2] Dept. of Computer Science and Engineering, University of South Carolina, , Columbia, SC, USA.

Abstract. We consider the problem of deploying robots to observe the evolution of a stochastic process in order to output a sequence of observations that fit some given specification. This problem often arises in contexts such as event reporting, situation depiction, and automated narrative generation. The paper extends our prior work by formulating and examining the multi-robot case: a team of robots move about, each recording what they observe, and, if they manage to capture some event, communicating that fact with the group. In the end, all events from all the robots are collated to provide a cumulative output. A plan is used to decide what each robot will attempt to capture next, based on the state of the world and the events that have been captured (collectively) so far. This paper focuses on the question of how to compute effective multi-robot plans. A monolithic treatment, involving the optimal selection of joint choices, i.e., choosing the next elements to attempt to capture by all robots, is formulated where costs are minimized in an expected sense. Since such plans are prohibitive to compute, variants based on an approximation scheme based on solving a sequence of individual planning problems are then introduced. This scheme sacrifices some solution quality but requires far less computational expense; we show this permits one to scale to greater numbers of robots.

Keywords: Robot videography, formal methods, heuristics for cooperative planning

1 Motivation

Imagine a nature documentary. Muffled, but in his signature rasping hush, David Attenborough intones: *"We see now the baby gazelle, utterly unaware of danger lurking close, as she edges toward the water's edge. Nearby, Mother gazelle is distracted, only for a moment, but..."* and the wild drama ensues—tooth, claw, and all. Later, as the credits roll by, it turns out that the rare footage making up

This material is based upon work supported by the National Science Foundation under Grants 1849249 & 1849291.

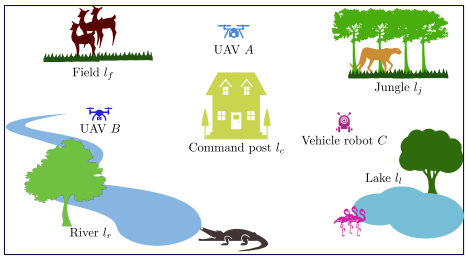


Fig. 1: A overall view of the wildlife reserve, with 5 main regions, a grass field l_f, a jungle l_j, a command post l_c, a riverside l_r, and a lakeside l_l. Notable fauna includes a cheetah, a crocodile, a herd of gazelle, and a flamboyance of flamingos; the latter two, being gregarious, remain as a group.

this documentary was captured not by expert human camera operators, but by a team of autonomous videographer robots. These robots, aided by traditional tags for tracking animals, have only a coarse sense of the locations of certain animals and are only able to make imperfect predictions for what activities the creatures will engage in. But they are also given a description of the sorts of events that are worth capturing, events to help describe Nature's unfolding story. Multiple robots ought to be engaged to ensure good coverage of the district, especially as there may be events of interest occurring simultaneously at different locations, such as when the flamingos take wing all at once, while a lone cheetah breaks cover from a thicket of trees elsewhere (See Figure 1). The robots plan their movements and capture events strategically so that, ultimately, the captured events form a depiction that is an engaging record of activity within the wildlife reserve. For example, one robot captures a majestic predator silhouetted against the moon, other robots capture (many) scenes of frolicking young. Comparatively little footage represents animals lolling about during the heat of the day.

This paper formalizes settings like the preceding scenario. Section 3 formulates the problem of coordinated story capture via: (*i*) a stochastic process to model worlds that generate random events, (*ii*) a means for expressing constraints on the robots' capabilities in terms of what they may capture in succession, and (*iii*) an automaton expressing sequences of observations (or recordings) to be captured. Next, in Section 4, we turn to finding plans for such problems: presenting both monolithic and decoupled (or sequentialized) approaches. Section 5, thereafter, returns to the nature documentary scenario to provide performance comparisons via case studies.

2 Related Work

Most closely related is our own prior work [1], which formalizes the problem of using autonomous robots to capture videos in unpredictable environments, an idea advanced in [2]. There, we studied the problem of computing, for a single robot, a policy that minimizes the expected number of steps to record an event sequence satisfying a given specification. This paper is a multi-agent extension of that the same problem, but with the objective being replaced by minimizing cost (rather than steps) of recording a desired event sequence. Here we also introduce a new structure, termed the *valid-action automaton*, which we use not only for specifying action-related costs but also for imposing spatial, temporal, or other constraints on the robots' choices of actions.

Also, some related problems bear similarity to robot video capture. Among them is the work of Yu and LaValle [3], who studied the *story validation problem*, the aim of which is to validate whether an event sequence captured by a set of sensors in the environment is consistent with a given story or not—this is, roughly, the inverse of the problem we consider here. The other two are the *video summarization problem* [4–9] and the *vacation snapshot problem* [10] the purpose of which are, respectively, to make a summary of a given video and to make a diverse selection of samples observed by a mobile robot. The essence in the video summarization problem is to post-processes a collection of images, while the idea in our problem is to decide which images the robot should attempt to capture without knowing which images will actually be realized by the world. For research about summarization in other contexts, see [11, 12] for generating commentary, and see [13, 14] for producing narrative text.

3 Definitions and Problem Statement

We start defining basic model elements, then give our formal problem statement.

3.1 Worlds and narratives: Event Model and Story Automaton

The atomic items that the robots capture are *events*, elements from a set E, which occur at specific times and places. The set of all event sequences (finite words) over E is denoted E^*. For integer m, the set of all event sequences over E with length at most m is denoted $E^{\leq m}$. We will mostly write sequences $e_1 e_2 \dots e_m$ of events, but occasionally it helps to treat them as tuples too, like $(e_1, e_2, \dots, e_m)$. For each $1 \leq i \leq m$, we will write $e_i \in (e_1, e_2, \dots, e_m)$, abusing notation to treat the tuple as a set as well. Using the tuple form, $E^* = \bigcup_{j=0}^{\infty} E^j$.

Events are assumed to be generated by a stateful stochastic process, unaffected by the actions of the robot.

Definition 1 (Event Model [1]). *An* event model $\mathcal{M} = (W, P, w_0, E, g)$ *has (1)* W*, which is a nonempty finite set, is the* state space *of the model; (2)* $P : W \times W \to [0, 1]$ *is the* transition probability function *of the model, such that for each state* $w \in W$*,* $\sum_{w' \in W} P(w, w') = 1$*; (3)* $w_0 \in W$ *is the* initial state*; (4)* E *is the set of all possible events; (5)* $g : W \times E \to [0, 1]$ *is a labeling function such that for each state* w *and event* e*,* $g(w, e)$ *is the probability that event* e *happens at state* w*. We assume that* $g(w_0, e) = 0$ *for any event* e*, meaning that no events happens at state* w*.*

The model assumes that the events in each state of the event model are both mutually and temporally independent. That is, the probability of occurrence of an event e in state w at time t does not depend on the probability of occurrence of any event e' at time t, nor at any time before that.

An execution of the system starts from w_0, and then at each time step t, the system transitions from state w_t to a state w_{t+1}, which is chosen randomly based on $P(w_t, \cdot)$. Accordingly, the system's execution goes through a path $w_0 w_1 \cdots$.

When the system enters state w_t, each event e occurs with probability $g(w_t, e)$. At a time step, it is possible that several events occur simultaneously.

In this paper, we consider a problem in which multiple robots are working cooperatively to record a sequence of events. As the system evolves along $w_0 w_1 w_2 \cdots$, the robots attempt to record some of the events that occur in the world to form a story $\xi \in E^*$. To specify the story, we use deterministic finite automaton (DFA) [15], called the **story automaton** $\mathcal{D} = (Q, E, \delta, q_0, F)$, in which Q is the state space, the set of events E is the alphabet, $\delta : Q \times E \to Q$ is the transition function, q_0 is the initial state, and $F \subseteq Q$ is the set of all accepting (final) states.

The language of the story automaton is denoted by $\mathcal{L}(\mathcal{D})$. Given a state $q \in Q$, we say that state q' is *reachable* from q by event sequence $e_1 e_2 \cdots e_m$ if there is a sequence of states $q_1, q_2, \ldots, q_{m+1}$ such that $q_1 = q$, $q_{m+1} = q'$, and $q_{i+1} = \delta(q_i, e_i)$ for each $1 \leq i \leq m$. We assume that each state is reachable from itself (by ϵ, the empty string).

3.2 Robot model

The current state of the event model is assumed to be observable to all the robots, i.e., at each time t, the robots know the current state of the event model (or the world) w_t; however, they do not know what the next state, w_{t+1}, will be. (In [1], our earlier work, we considered the case where the current state of the event model was not observable: for simplicity, especially in exploring the complexities arising in the multi-robot case, the present paper utilizes MDPs throughout, rather than the less tractable POMDP model.)

Generally, robots use actions to alter their relationship with the world—here, the robots also attempt to record events. To cooperatively capture a sequence of events each of the n robots chooses an action to execute from A, the set of all possible actions. Each action is associated with the event they aim to capture via the *recording function*, $r : A \to E \cup \{\epsilon\}$. Since some actions may not involve any recording, the symbol ϵ is included, indicating that no event will ever be captured by the associated action. At every time step, each robot executes an action from A, if that action aims to capture an event and that event occurs during the execution of the action, the robot will succeed in capturing that event. We assume every event can be recorded by some action, i.e., for every $e \in E$, there is some action $a_e \in A$ such that $r(a_e) = e$. Further, the set of actions includes a *no-action* choice, $\perp \in A$, that does nothing and records no event, $r(\perp) = \epsilon$. Occasionally we will apply $r(\cdot)$ to a tuple in a point-wise fashion.

Owing to constraints, present either in the world or in the way the robots interact with the world, not all actions can be executed at all times. Hence, an action $a \in A$ with $r(a) = \epsilon$ may still be useful because, though it won't capture an event itself, it may alter what can be captured subsequently. Think, for instance, of a robot using the time step's duration to shift location, or to deploy a stalking horse. The following structure expresses such constraints and also associates costs to each action.

Definition 2 (Valid-action Automaton (VA)). *For robot $i \in \{1, \ldots, n\}$, we define its* valid-action automaton *as a 5-tuple, $\mathbb{V}^{(i)} = (V^{(i)}, v_0^{(i)}, A, \tau^{(i)}, c^{(i)})$, (1) $V^{(i)}$ is the set of vertices; (2) $v_0^{(i)} \in V^{(i)}$ is the initial vertex; (3) A, its alphabet, a set of all possible robot actions; (4) $\tau^{(i)} : V^{(i)} \times A \hookrightarrow V^{(i)}$, which could be partial, is the transition function; (5) $c^{(i)} : V^{(i)} \times A \to \mathbb{R}_{>0} \cup \{+\infty\}$ is the cost function, such that for each $(v, a) \in V^{(i)} \times A$, $c^{(i)}(v, a)$ is the cost of taking action a at vertex v. We assume that $c^{(i)}(v, a) = +\infty$ for any (v, a) such that $\tau^{(i)}(v, a)$ is not defined.*

Each robot $i \in \{1, 2, \ldots, n\}$ keeps track of the current state of its own valid-action automaton, denoted $v_t^{(i)}$. Actions are performed as follows. Robot i makes a choice, from among those actions a for which $\tau(v_t^{(i)}, a)$ is defined, to enact at time $t + 1$. We denote the action $a_t^{(i)}$, because it is chosen at time t. The world evolves from w_t to w_{t+1}, and robot i pays cost $c^{(i)}(v_t^{(i)}, a_t^{(i)})$ executing $a_t^{(i)}$ to change its circumstances, with aspects relevant for subsequent actions being represented in $v_{t+1}^{(i)}$. Finally, if $r(a_t^{(i)}) \neq \epsilon$, the robot attempts to record event $r(a_t^{(i)}) \in E$, which succeeds with probability $g(w_{t+1}, r(a_t^{(i)}))$.

For each time step $t \geq 1$, we define $X_t \subseteq E^{\leq n}$ to be the set of all event sequences, in any order, formed from all the events that were captured by the robots at time step t. For example, if at time step t_0, e_1 was captured by robot 1, e_2 was captured by robot 2, and ϵ (nothing) was captured by robot 3, then $X_{t_0} = \{e_1e_2, e_2e_1\}$. We also let $\mathbf{X}_t = \prod_{i=1}^{t} X_i$ be the set of all event sequences obtained by concatenating the event sequences made for the time steps $1, \ldots, t$. As an example, if $X_1 = \{e_3, e_4e_2\}$ and $X_2 = \{e_1e_2, e_2e_1\}$, then $\mathbf{X}_2 = \{e_3e_1e_2, e_3e_2e_1, e_4e_2e_1e_2, e_4e_2e_2e_1\}$. The robots check at each time t, if there exists an event sequence $x \in \mathbf{X}_t$ such that $x \in \mathcal{L}(\mathcal{D})$ or not. If yes, then it means that the robots have successfully collected events to make a desired story, namely x, and they terminate. Note that $\mathbf{X}_t$ is the set of all event sequences the robots can make by concatenating all the events they have captured until time step t with the constraint that for times t_1 and t_2 for which $t_1 < t_2$, no event captured at t_2 precedes an event captured at t_1.

3.3 Policies and Problem Statement

The robots' choice of actions is governed by a policy $\pi(\cdot, \cdot, \cdot)$, that, at time t, based on the current state of the event model, states in the story automaton, and current states in the robots' valid-action automata, produces a n-tuple of actions, termed a joint action, telling each robot what action to execute.

Given valid-action automata $\mathbb{V}^{(i)} = (V^{(i)}, v_0^{(i)}, A, \tau^{(i)}, c^{(i)})$, $i \in \{1, \ldots, n\}$, we will write $\mathbf{V} = V^{(1)} \times \cdots \times V^{(n)}$. Similarly, for joint actions, we have $\mathbf{A} = A \times \cdots \times A = A^n$. (To lighten the notation, we assume that A is identical for every robot; no generality is lost because, should robot i be unable to execute some $a \in A$, then a simply does not appear in $\mathbb{V}^{(i)}$.)

Given $\mathbb{V}^{(i)}$ for $i \in \{1, \ldots, n\}$, we define $\mathbf{c} : \mathbf{V} \times \mathbf{A} \to \mathbb{R}_{>0}$, the *aggregate cost function*, by $\mathbf{c}((v^{(1)}, \ldots, v^{(n)}), (a_1, \ldots, a_n)) = \sum_{i=1}^{n} c^{(i)}(v^{(i)}, a_i)$. Then the total

cost incurred up until time t is $J_t = \sum_{i=0}^{t-1} \mathbf{c}((v_t^{(1)}, \ldots, v_t^{(n)}), (a_t^{(1)}, \ldots, a_t^{(n)}))$. Let T be the first time such that $\mathbf{X}_T \cap \mathcal{L}(\mathcal{D}) \neq \varnothing$, then we say the story has been captured at time T and we write the cost of capturing the story as $J = J_T$.

We now define the problem we study in this paper.

Problem: Multi-Robot Recording Cost Minimization (MRRCM)

Input: An event set E; an event model $\mathcal{M} = (W, P, w_0, E, g)$; the story automaton $\mathcal{D} = (Q, E, \delta, q_0, F)$; the number of robots, n; a set of n valid-action automata $\mathbb{V}^{(i)} = (V^{(i)}, v_0^{(i)}, A, \tau^{(i)}, c^{(i)})$, $\forall i \in \{1, \cdots n\}$.

Output: A policy, $\pi^* : S \times 2^Q \times \mathbf{V} \to \mathbf{A}$, that minimize the expected cost, J, to capture a story in $\mathcal{L}(\mathcal{D})$.

4 Solving MRRCM

In this section we provide two algorithms to solve MRRCM.

4.1 Preliminary definitions

In the initial step, the algorithm makes use of the story automaton and n, the number of robots, to construct a *footage automaton* as follows:

Definition 3 (Footage Automaton). *Let $\mathcal{D} = (Q, E, \delta, q_0, F)$ be the story automaton and n be the number of robots. We construct the* footage automaton *as a nondeterministic finite automaton (NFA) $\mathbf{N} = (Q, \mathbf{E}, \delta_{\mathbf{N}}, q_0, F)$, where (1) Q is the state space; (2) $\mathbf{E} = \{(e^{(1)}, \ldots, e^{(n)}) \mid e^{(i)} \in E \cup \{\epsilon\}, \forall i \in \{1, \ldots, n\}\}$ is its alphabet; (3) $\delta_{\mathbf{N}} : Q \times \mathbf{E} \hookrightarrow 2^Q$, is the transition function, such that for $q \in Q$ and $(e^{(1)}, \ldots, e^{(n)}) \in \mathbf{E}$, $\delta_{\mathbf{N}}(q, (e^{(1)}, \ldots, e^{(n)})) = \{q_i | q_i \in Q$, where, in $\mathcal{D}$, q_i is reachable from q using some permutation of the tuple $(e^{(1)}, \ldots, e^{(n)})\}$; (4) q_0 is the initial state; (5) F is the set of final states.*

Each transition starting from a state in the footage automaton corresponds to at most n consecutive transitions starting from that state in the story automaton. The idea is that the footage automaton tracks the story automaton states which can be reached using the events captured by all of the robots. The next step converts the footage automaton $\mathbf{N}$ into a *deterministic footage automaton* $\mathbf{D} = (\mathbf{Q}, \mathbf{E}, \delta_{\mathbf{D}}, q_0, \mathbf{F})$, which is, in fact, a deterministic finite automaton, using the well-known technique of NFA to DFA conversion [15]. The number of edges in the constructed $\mathbf{N}$, of the output $\mathbf{D}$, and the work needed in this conversion step, can be reduced by fixing a canonical representative, equivalent up to permutation, for the n-tuples comprising $\mathbf{E}$. Sorting the tuples works.

We define a function $h : \mathbf{A} \times 2^E \to (E \cup \{\epsilon\})^n$ such that for each joint action $\mathbf{a} \in \mathbf{A}$, and a set of events $B \subseteq E$, $h(\mathbf{a}, B) = (d^{(1)}, \ldots, d^{(n)})$ in which for each $j \in \{1, \ldots, n\}$, $d^{(j)} = r(a^{(j)})$ if $r(a^{(j)}) \in B$, otherwise $d^{(j)} = \epsilon$.[‡] Intuitively, given

[‡]Being consistent with that above, we use $a^{(j)}$ to denote the jth element of $\mathbf{a}$.

that only the events in B happen, the function h outputs an n-tuple of events which are captured by the action $\mathbf{a}$. We then define $o : \mathbf{A} \to 2^{(E \cup \{\epsilon\})^n}$ such that for each $\mathbf{a} \in \mathbf{A}$, $o(\mathbf{a}) = \bigcup_{B \subseteq E} \{h(\mathbf{a}, B)\}$. This function produces any tuple of events that could be captured by a joint action.

Two additional functions will be needed. Let $\varrho : \mathbf{A} \times W \times (E \cup \{\epsilon\})^n \hookrightarrow \mathbb{R}_{\geq 0}$ be a function such that for each $\mathbf{a} \in \mathbf{A}$, $w \in W$, and $\mathbf{b} \in o(\mathbf{a})$,

$$\varrho(\mathbf{a}, w, \mathbf{b}) = \Big(\prod_{e_0 \in \mathbf{b}} g(w, e_0) \Big) \cdot \Big(\prod_{\substack{e_1 \in r(\mathbf{a}) \\ e_1 \notin \mathbf{b}}} (1 - g(w, e_1)) \Big).$$

The interpretation is: assuming that at time t the robots execute joint action $\mathbf{a}$ and, at $t+1$, the event model transitions to w, then $\varrho(w, \mathbf{a}, \mathbf{b})$ gives the probability that $\mathbf{b}$ is realized by w. Or, in other words, among those events attempted to be captured by $\mathbf{a}$, only those within $\mathbf{b}$ happened in state w.

Next, using $\mathbb{1}_A(\cdot)$ for set A's indicator function, let $\lambda : \mathbf{Q} \times \mathbf{A} \times W \times \mathbf{Q}$ be

$$\lambda(\mathbf{q}, \mathbf{a}, w, \mathbf{q}') = \sum_{\mathbf{b} \in o(\mathbf{a})} \mathbb{1}_{\{\mathbf{q}'\}}(\delta_{\mathbf{D}}(\mathbf{q}, \mathbf{b})) \cdot \varrho(\mathbf{a}, w, \mathbf{b}).$$

At time t, if the footage automaton is in state $\mathbf{q}$ and the robots execute $\mathbf{a}$, and thereupon the event model transitions next to state w, then $\lambda(\mathbf{q}, \mathbf{a}, w, \mathbf{q}')$ is the probability that the footage automaton transitions to $\mathbf{q}'$ at $t+1$.

With these definitions, we now present our algorithms for solving MRRCM.

4.2 Full Joint Plan

The first step of the algorithm makes from valid-action automata of the robots, an automaton defined as follows:

Definition 4 (Joint Action Automaton (JA)). *Given valid-action automata* $\mathbb{V}^{(i)} = (V^{(i)}, v_0^{(i)}, A, \tau^{(i)}, c^{(i)})$ *for* $i \in \{1, 2, \ldots n\}$*, and the aggregate cost function* $\mathbf{c} : \mathbf{V} \times \mathbf{A} \to \mathbb{R}_{>0}$*, their* joint action automaton *is* $\mathcal{V} = (\mathbf{V}, \mathbf{v_0}, \mathbf{A}, T, \mathbf{c})$*, where: (1)* $\mathbf{V} = V^{(1)} \times V^{(2)} \times \cdots \times V^{(n)}$ *is the set of all the vertices; (2)* $\mathbf{v_0} = (v_0^{(1)}, v_0^{(2)}, \ldots, v_0^{(n)})$ *is the initial vertex; (3)* $\mathbf{A} = A^n$ *is the set of all actions; (4)* $T : \mathbf{V} \times \mathbf{A} \hookrightarrow \mathbf{V}$ *is the valid transitions function, such that for each* $(v^{(1)}, v^{(2)}, \ldots, v^{(n)}), (w^{(1)}, w^{(2)}, \ldots, w^{(n)}) \in \mathbf{V}$ *and* $(a^{(1)}, a^{(2)}, \ldots, a^{(n)}) \in \mathbf{A}$*,* $T((v^{(1)}, v^{(2)}, \ldots, v^{(n)}), (a^{(1)}, a^{(2)}, \ldots, a^{(n)})) = (w^{(1)}, w^{(2)}, \ldots, w^{(n)})$ *if for each* $i \in \{1, \ldots, n\}$*,* $\tau^{(i)}(v^{(i)}, a^{(i)}) = w^{(i)}$*; (5)* $\mathbf{c} : \mathbf{V} \times \mathbf{A} \to \mathbb{R}_{>0}$ *is the aggregate cost function.*

Now, to solve the MRRCM problem jointly for all the robots, we search over all a joint action space. To do so we construct an MDP, called the *joint MDP*.

Definition 5 (Joint MDP). *For event set* E *and model* $\mathcal{M} = (W, P, w_0, E, g)$*, joint action automaton* $\mathcal{V} = (\mathbf{V}, \mathbf{v_0}, \mathbf{A}, T, \mathbf{c})$*, and the deterministic footage automaton* $\mathbf{D} = (\mathbf{Q}, \mathbf{E}, \delta_{\mathbf{D}}, q_0, \mathbf{F})$*, construct* $\mathbb{M}_{\mathcal{M},\mathbf{D},\mathcal{V}} = (\mathbf{S}, s_0, \mathbb{A}, \mathbf{P}, \mathbf{J}, \mathbf{G})$*, where (1)* $\mathbf{S} \subseteq W \times \mathbf{Q} \times \mathbf{V}$*, is the set of states; (2)* $\mathbf{s_0} = (w_0, q_0, \mathbf{v_0}) \in \mathbf{S}$ *is the*

initial state; (3) $\mathbb{A} : \mathbf{S} \to 2^{\mathbf{A}}$ *is the action function, such that for each* $\mathbf{s} = (w, \mathbf{q}, \mathbf{v}) \in \mathbf{S}$, $\mathbb{A}(\mathbf{s}) = \{\mathbf{a} \in \mathbf{A} \mid \delta_{\mathbf{D}}(\mathbf{q}, r(\mathbf{a}))$ *and* $T(\mathbf{v}, \mathbf{a})$ *are defined*$\}$*; (4)* $\mathbf{P} : \mathbf{S} \times \mathbf{A} \times \mathbf{S} \to [0, 1]$*, is the probability function,* $\mathbf{P}((w, \mathbf{q}, \mathbf{v}), \mathbf{a}, (w', \mathbf{q}', \mathbf{v}')) = \mathbb{1}_{\{\mathbf{v}'\}}(\mathbf{T}(\mathbf{v}, \mathbf{a})) \cdot P(w, w') \cdot \lambda(\mathbf{q}, \mathbf{a}, w', \mathbf{q}')$*; (5)* $\mathbf{J} : \mathbf{S} \times \mathbf{A} \to \mathbb{R}$*, is the cost function,* $\mathbf{J}((w, \mathbf{q}, \mathbf{v}), \mathbf{a}) = (1 - \mathbb{1}_{\mathbf{F}}(\mathbf{q})) \cdot \mathbf{c}(\mathbf{v}, \mathbf{a})$*; (6)* $\mathbf{G} = W \times \mathbf{F} \times \mathbf{V}$ *is the set of goal states.*

Note that this construction is a goal MDP, meaning it is an MDP supplemented with a set of goal states. The MRRCM problem then is reduced to finding for this MDP, a policy that minimizes the expected cost of reaching goal states. Such a policy, denoted $\pi^*_{\mathbb{M}}$, is a function $\pi^*_{\mathbb{M}} : \mathbf{S} \to \mathbf{A}$ which gives the optimal action for each $\mathbf{s} \in \mathbf{S} \setminus \mathbf{G}$ using the Bellman equation, which may be computed by a variety of methods [16].

Note that for all $\mathbf{s} \in \mathbf{G}$, $V^*_{\mathbb{M}}(\mathbf{s}) = 0$. As each state $s \in \mathbf{S}$ is a triple $(w, \mathbf{q}, \mathbf{v})$, and each state $\mathbf{q}$ of the deterministic footage automaton corresponds to a set of states of the story automaton, the computed policy $\pi^*_{\mathbb{M}}$ is a solution to MRRCM.

This MDP has a state space of size $\Theta(|W||\mathbf{Q}||V^{(1)}||V^{(2)}| \cdots |V^n|)$ and an action space of size $|A|^n$. As the number of robots increases, both the state space and the action space of the MDP grows exponentially. Because of this, the next section pursues a solution to the MRRCM problem with less expense.

4.3 Sequentialized Planning

The overall idea of our second algorithm, following the classical approach in multi-agent planning [17], is to solve a sequence of MDPs, each being considerably smaller than the full joint MDP. Each MDP is constructed for a single robot in the team, the structure of each conditioned on the optimal policies of those preceding it in the sequence. Each is a goal MDP and, hence, an optimal policy can be computed via Bellman recurrences.

Suppose that the n robots are ordered: $j_1 j_2 \ldots j_n$, where $j_k \in \{1, \ldots, n\}$. If $j_1, \ldots, j_{k-1}$ have determined how they will act, robot j_k can solve an MDP with stochastic transitions incorporating the events that the other $k-1$ might record, along with the associated probabilities of the events actually occurring, as gratis contributions. Once robot j_k solves this to obtain a policy, we have policies for the first k robots, and could proceed onward to robot j_{k+1}. And so on, until j_n.

The difficulty is that, even if the $k-1$ robots do have policies, those policies involve states within $\mathbb{V}^{(j_1)}, \mathbb{V}^{(j_2)}, \ldots, \mathbb{V}^{(j_{k-1})}$, which is information that robot j_k is not privy to, so policies will not give a determination of the actions of the first $k-1$ robots. Even if the robot had that information—obtained, say, by copious broadcast communication—this would still yield a policy for j_k as a function over $W \times \mathbf{Q} \times V^{(j_1)} \times \cdots V^{(j_k)}$, which grows exponentially in n in the worse case.

We pursue the following alternative, with more attractive scaling properties. For robot j_k, we compute a policy over state space $W \times \mathbf{Q} \times V^{(j_k)}$. The construction of the j_k's MDP is as follows:

Definition 6. *For robot* j_k*, event model* $\mathcal{M} = (W, P, w_0, E, g)$*, deterministic footage automaton* $\mathbf{D} = (\mathbf{Q}, \mathbf{E}, \delta_{\mathbf{D}}, q_0, \mathbf{F})$*, valid-action automaton for the*

robot $\mathbb{V}^{(j_k)} = (V^{(j_k)}, v_0^{(j_k)}, A, \tau^{(j_k)}, c^{(j_k)})$*, policies* π_{j_m} *for* $m \in \{1, \ldots, k-1\}$*, and a distribution over the valid-action automata states for the* $k-1$ *robots,* $\Delta^{(j_m)} : V^{(j_m)} \to [0,1]$ *for* $m \in \{1, \ldots, k-1\}$*, we construct the sequential MDP* $\mathbb{M}_{j_k} = (S^{(j_k)}, s_0^{(j_k)}, A, \mathbf{P}^{(j_k)}, J^{(j_k)})$*, where*

- $S^{(j_k)} \subseteq W \times \mathbf{Q} \times V^{(j_k)}$ *is the state space;*
- $s_0^{(j_k)} = (w_0, q_0, v_0^{(j_k)}) \in S^{(j_k)}$ *is the initial state;*
- A *is the action space;*
- $\mathbf{P}^{(j_k)} : S^{(j_k)} \times A \times S^{(j_k)} \to [0,1]$ *is the probability function such that*

$$\mathbf{P}^{(j_k)}(s, a, s') = \mathbb{1}_{\{v'\}}(\tau^{(j_k)}(v, a)) \sum_{\boldsymbol{\alpha} \in \mathbf{A}_a^k} P(w, w') \mu_{w,\mathbf{q}}(\boldsymbol{\alpha}) \lambda(\mathbf{q}, \boldsymbol{\alpha}, w', \mathbf{q}')$$

  with $s = (w, \mathbf{q}, v)$*,* $s' = (w', \mathbf{q}', v')$*, and where* $\mu_{w,\mathbf{q}} : \mathbf{A}_a^k \to [0,1]$ *is*

$$\mu_{w,\mathbf{q}}(\boldsymbol{\alpha}) = \sum_{\substack{v^{(1)} \in V^{(j_1)} \\ \vdots \\ v^{(k-1)} \in V^{(j_{k-1})}}} \prod_{m=1}^{k-1} \left(\mathbb{1}_{\{a^{(m)}\}}(\pi_{j_m}(w, \mathbf{q}, v^{(m)})) \Delta^{(j_m)}(v^{(m)}) \right)$$

  and $\mathbf{A}_a^k$ *consists of joint actions* $(a^{(1)}, a^{(2)}, \ldots, a^{(k-1)}, a, \overbrace{\perp, \perp, \cdots, \perp}^{n-k})$*;*
- $J^{(j_k)} : S^{(j_k)} \times A \to \mathbb{R}$ *is the cost function, such that for* $s = (w, \mathbf{q}, v) \in S^{(j_k)}$ *and* $a \in A$*, we have* $J^{(j_k)}(s, a) = (1 - \mathbb{1}_{\mathbf{F}}(\mathbf{q}))\, c^{(j_k)}(v, a)$*.*

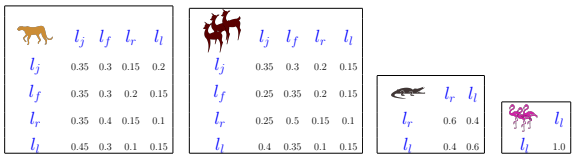
The intuition here is that, in lieu of actual information on the state of each $\mathbb{V}^{(j_k)}$, estimates (in the form of distribution $\Delta^{(j_k)}$) are used as an approximation. In what follows, we make a maximum entropy assumption over the states of the valid-action automata, i.e., $\Delta^{(i)}(v) = \frac{1}{|V^{(i)}|}$, though cleverer choices exist.

Based on this treatment, one expects that the ordering of the robots would affect the overall solution quality. Though a random order will work, it may fail to give a good policy so we employ the following greedy heuristic to choose a favourable ordering. First, we calculate the policies for all n robots tentatively assuming each would be operating alone. Then we select the robot whose individual policy gives the least expected cost to capture the story, and use it as the robot for the first spot. Having determined j_1, we compute policies for the remaining $n-1$ robots, given j_1 and its policy π_{j_1}. The robot with the least expected cost becomes j_2, and the process is repeated but now with $\{j_1, j_2\}$ determined. This is repeated until all n have been ordered.

5 Case Study

In this section, we present results of our Python implementation of the algorithms, which we executed on an Ubuntu 16.04 computer with a 3.6GHz CPU.

We revisit the shooting of documentary in a wildlife reserve as used as motivation initially, and outlined in Figure 1. A system-level event model is constructed from event models for the animals. Figure 2 shows, for each type of animal, the transition probability function P of its event model, in which each entry $P(l_1, l_2)$



	l_j	l_f	l_r	l_l
l_j	0.35	0.3	0.15	0.2
l_f	0.35	0.3	0.2	0.15
l_r	0.35	0.4	0.15	0.1
l_l	0.45	0.3	0.1	0.15

	l_j	l_f	l_r	l_l
l_j	0.35	0.3	0.2	0.15
l_f	0.25	0.35	0.2	0.15
l_r	0.25	0.5	0.15	0.1
l_l	0.4	0.35	0.1	0.15

	l_r	l_l
l_r	0.6	0.4
l_l	0.4	0.6

	l_l
l_l	1.0

Fig. 2: Transition probabilities of the animals.

is the probability that the animal(s) go from the current location l_1, to l_2 in the next hour. Based on these event models, we see the flamingos are only interested in the lake, while the crocodile is interested in both the river and the lake. The others roam more widely. The event model for the whole system is obtained via a Cartesian product of these individual models.

The events of interest are gazelle grazing, g_e; cheetah eating gazelle, c_g; crocodile eating gazelle, k_g; flamingos mating, f_m; and crocodile eating flamingo, k_f. Event g_e happens with probability 1 for both cases, whether the gazelle are in the field or the jungle. The probabilities that the cheetah can successfully capture and kill a gazelle in the field and the jungle are, respectively, 0.2 and 0.25. The probabilities that the crocodile can successfully capture and kill a gazelle by the river and by the lake are, respectively, 0.2 and 0.25. The probability that the crocodile can capture and kill a flamingo is 0.15. The probability that a flamingos mating event happens in an hour is 0.4.

The robots cannot track or follow animal(s) continually, and video clips are only recorded at locations l_f, l_j, l_r, and l_l. At each time step, each robot observes the current state of the event model by receiving a message from the command post, which reports the rough locations of the animals based on GPS trackers connected to tags on the animals. The robot does not know which location will be the creatures' destination in the next hour. Also, at every step t, each robot relays to the other robots whether it was successful in recording the event it attempted. Accordingly, all the robots can compute $\mathbf{X}_t$ and $\mathbf{Q}_t$. At each time step, a robot must guess the destination of an animal of interest so that it can go to that destination and prepare to capture a desired event.

Three kinds of videographer robots A, B, and C are available. The robots are camouflaged, so their presence does not change the animals' behavior, i.e., we make a non-causality assumption. Robots of type A are UAVs capable of traveling between any of those locations in a time step. Its actions are unconstrained, so it has a trivial (single state) valid-action automaton. Robots of type B are short-endurance UAVs with a simple constraint on their actions: if at time t, such a robot chooses event k_g to capture at time $t+1$, then it is forbidden from repeating it (regardless of whether it successfully captured k_g at $t+1$ or not). Robots of type C are ground vehicles. When starting out from the river or lake, these vehicles must take a detour, visiting the command post (for one time step), to be outfitted with equipment required to travel through the jungle or field. Likewise, when in the jungle/field, travel to riverside/lakeside, requires a sojourn at the command post first. Accordingly, if a robot of type C moves to l_c, then in the next time step no event will be captured by that robot.

We set the cost function in a way that the objective in the MRRCM problem becomes minimizing the expected number of hours to capture a desired story.

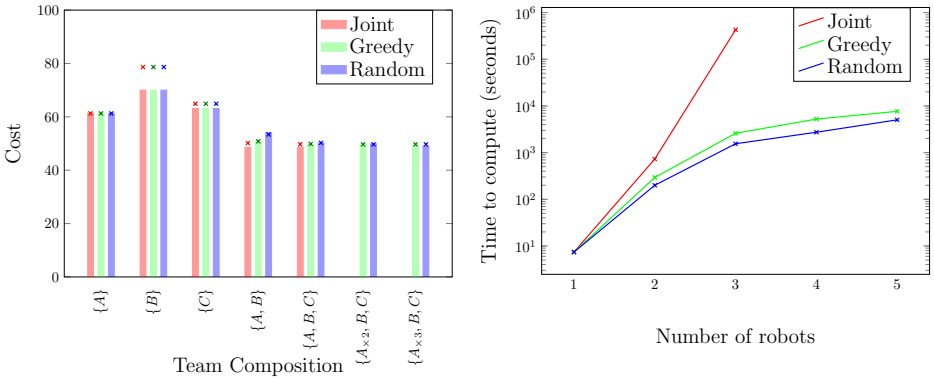


Fig. 3: Left: Cost to capture story. The theoretical prediction (expected cost for policy) is shown via × marks. Average cost for 1000 simulation is shown via bars. Right: Time to compute the policies. (The y-axis is in the logarithmic scale).

For this purpose, for each robot i and action a, for each vertex v of the valid-action automaton of robot i, we set $c^{(i)}(v, a) = 1$. Because the joint cost of a group of robots is the sum of costs for individual robots, in reporting the results we have divided by the number of robots, each of the expected and the average costs obtained for a joint plan so that these figures represent, respectively, the excepted number of hours and the average number of hours to record a story. For the sequential plan, no division is needed and we use the expected and average costs obtained for the last MDP in the sequential plan directly.

In this case study we are interested in capturing a sequence that is a supersequence for both the sequences $g_e f_m c_g k_f$ and $g_e f_m k_g k_f$, each of which chronicles both a gazelle's life and a flamingo life. Once a desired sequence was captured, we post-process it to make two videos $g_e f_m c_g k_f$ and $g_e f_m k_g k_f$ from it, each for a TV channel. Note that the language of the specification DFA in this case is infinite. We considered several scenarios in which different numbers of each type of robot are tasked to capture a desired story. For our implementations, we use the value-iteration method to solve the underlying MDPs. For each scenario, we solved the MRRCM problem using the joint approach and the sequential approach with the random and greedy strategies. Also for each scenario, we generated 1000 simulations of executing the event model and for each simulation the robot(s) use the computed policy to capture a story. For each case, we computed the average cost of capturing a desired story over the 1000 simulations. Figure 3 shows the expected number of hours and the average number of hours for those simulations. As it was expected, a robot of type A outperformed the other two robot types B and C in yielding a smaller expected cost. Also for each experiment, the expected cost was very close to the average cost for 1000 simulations. We were able to generate a joint plan only for up to three robots; it took approximately 14 hours to generate a joint plan for three robots, while generating a sequential plan for three robots with each of the random and the greedy strategies took approximately 43 minutes and 85 minutes respectively.

6 Conclusion

This paper considered the problem of computing a policy, for a team of robots, minimizing the expected cost of recording a sequence of events that happen unpredictably. The problem is reduced to that of computing an optimal policy for a joint MDP of the robots. To overcome the computational complexities of solving the joint MDP, we proposed to solve the problem via a sequence of MDPs, using a greedy heuristic to order that sequence, and finally we presented our implementation results via a wild life case study.

References

1. H. Rahmani, D. A. Shell, and J. M. O'Kane, "Planning to chronicle," in *Workshop on the Algorithmic Foundations of Robotics (WAFR XIV)*, 2020.
2. D. A. Shell, L. Huang, A. T. Becker, and J. M. O'Kane, "Planning coordinated event observation for structured narratives," in *IEEE ICRA*, 2019.
3. J. Yu and S. M. LaValle, "Story validation and approximate path inference with a sparse network of heterogeneous sensors," in *IEEE ICRA*, 2011.
4. Y. J. Lee, J. Ghosh, and K. Grauman, "Discovering important people and objects for egocentric video summarization," in *IEEE CVPR*, 2012.
5. R. Hong, J. Tang, H.-K. Tan, C.-W. Ngo, S. Yan, and T.-S. Chua, "Beyond search: Event-driven summarization for web videos," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 7, no. 4, p. 35, 2011.
6. D. Potapov, M. Douze, Z. Harchaoui, and C. Schmid, "Category-specific video summarization," in *ECCV*, 2014.
7. L. Feng, Z. Li, Z. Kuang, and W. Zhang, "Extractive video summarizer with memory augmented neural networks," in *ACM Multimedia*, 2018.
8. P. Chang, M. Han, and Y. Gong, "Extract highlights from baseball game video with hidden markov models," in *IEEE ICIP*, 2002.
9. M. H. Kolekar and S. Sengupta, "Event-importance based customized and automatic cricket highlight generation," in *IEEE ICME*, 2006.
10. Y. Girdhar, P. Giguere, and G. Dudek, "Autonomous adaptive exploration using realtime online spatiotemporal topic modeling," *International Journal of Robotics Research*, vol. 33, no. 4, pp. 645–657, 2014.
11. H. Hajishirzi, J. Hockenmaier, E. T. Mueller, and E. Amir, "Reasoning in Robocup Soccer Narratives," in *UAI*, 2011.
12. S. Rosenthal, S. P. Selvaraj, and M. Veloso, "Verbalization: Narration of autonomous robot experience," in *IJCAI*, 2016.
13. M. O. Riedl and R. M. Young, "Narrative Planning: Balancing Plot and Character," *Journal of Artificial Intelligence Research*, vol. 39, pp. 217–268, 2010.
14. C. Barot, M. Branon, R. E. Cardona-Rivera, M. Eger, M. Glatz, N. Green, J. Mattice, C. Potts, J. Robertson, M. Shukonobe, L. Tateosian, B. R. Thorne, and R. M. Young, "Bardic: Generating Multimedia Narrative Reports for Game Logs," Working Notes of the AIIDE Workshop on Intelligent Narrative Technologies, 2017.
15. J. E. Hopcroft and J. D. Ullman, *Introduction to Automata Theory, Languages and Computation. Adison-Wesley.* Reading, Mass, 1979.
16. M. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming.* John Wiley and Sons, 1994.
17. E. Ephrati and J. S. Rosenschein, "Divide and conquer in multi-agent planning," in *AAAI-94*, 1994.