Bayesian Networks for Path-Based Sensors: Gathering Information and Path Planning in Communication Denied Environments

Alkesh K. Srivastava[1,2], George P. Kontoudis[1,3], Donald Sofge[1,4], and Michael Otte[1]

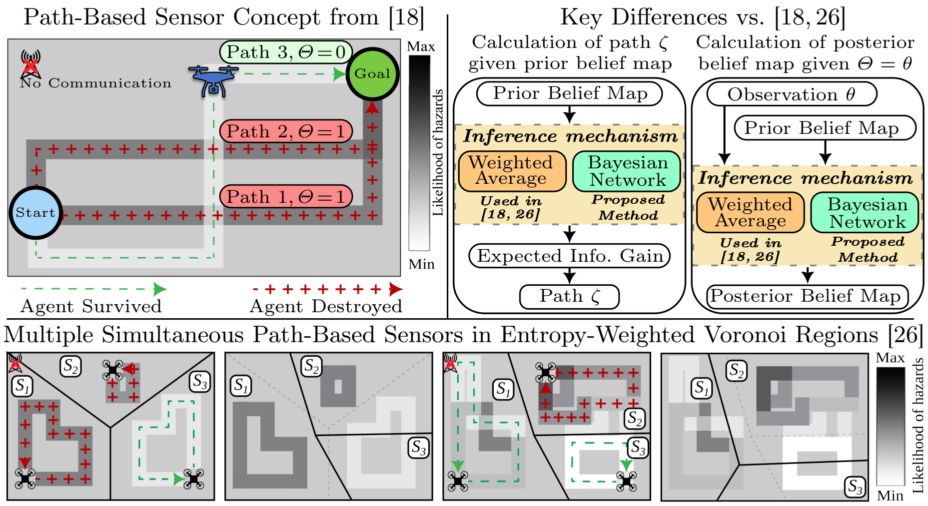
[1] University of Maryland, College Park, MD, US. otte@umd.edu
[2] Temple University, Philadelphia, PA, US. alkesh@temple.edu
[3] Colorado School of Mines, Golden, CO, US. george.kontoudis@mines.edu
[4] U.S. Naval Research Lab (Retired), DC, US. donald.sofge@nrl.navy.mil

Abstract. A "path-based sensor" *produces a single observation* along a continuous path. For example, a boolean path-based sensor returns a single "1" if an event of interest is detected at any point along the path and a "0" otherwise. Notably, a "1" provides no direct information about where along the path the event(s) may have occurred. Previous work has demonstrated that observations from multiple path-based sensors can be fused to create a Bayesian belief map over the spatial locations of the underlying event or phenomenon. Moreover, path planning can employ Shannon information theory to accelerate the rate of convergence of the belief map. In this paper, we present a new method to update the belief map based on a path-based sensor observation, and then plan paths to increase information gain. In contrast to prior work that approximates the posterior by averaging over the alternative event histories, we introduce a Bayesian Network (BN) formulation that models the probabilistic relationships between the latent variables and path-based sensor measurements, enabling a more principled Bayesian belief update. We consider static hazard detection in a communication-denied environment as a representative problem setting. The event of a robot returning from its path corresponds to a path-based hazard sensors reading of "0" (hazard *not* detected) while a robot failing to return corresponds to a reading of "1" (hazard detected). We consider false positives and false negatives. We find that the new method leads to quicker convergence of the belief map versus prior work in both single- and multi-robot cases.

Keywords: Path Planning · Multi-Agent Systems · Bayesian Inference · Shannon Information · Target Search · Path-Based Sensing.

1 Introduction

Unlike conventional sensors, which take measurements or observations from a single point or a sensor footprint, a *path-based sensor* takes a single measurement or observation along a trajectory. A boolean path-based sensor returns



(a) Agents plan and traverse paths in their search spaces.

(b) New computation of entropy-weighted Voronoi partitions

(c) Agents plan and traverse paths in their search spaces.

(d) New computation of entropy-weighted Voronoi partitions

Fig. 1. Top-Left: Belief about hazard existence (grayscale) from three path-based sensor observations. Agents are destroyed along paths 1 and 2 (red), causing two sensor triggers ($\Theta = 1$) that increase belief about hazards along those paths (dark gray). An agent survives path 3 (light green) causing a non-trigger ($\Theta = 0$) decreasing belief about hazards (light gray). **Bottom:** Extension of this idea to multiple simultaneous agent deployment, each in its own entropy-weighted Voronoi partition. **Top-Right:** Conceptual framework highlighting the key difference between the approach presented in this paper using a Bayesian Network (green) versus prior work [18, 26] (orange).

a "1" or "0," respectively, depending on whether or not an event of interest is observed (at least once) along an entire trajectory. As an illustrative example, consider an autonomous underwater robot that slowly collects water in a tank as it travels between a start and goal location. After the mission, the collected water is analyzed in a laboratory to determine whether a particular microorganism is present. While it is possible to determine (after the analysis) that one or more microorganisms have been collected, the exact location(s) of collection are not directly observable. Assuming that the habitats of the microorganisms are known to be localized to a small number of (initially unknown) regions within the environment, we can use multiple robot deployments and laboratory tests to identify where the habitat regions are located. In particular, we can refine a Bayesian belief map of microorganism habitat across the environment while deploying our robot (or even multiple robots) in an iterative fashion. Moreover, if the robot's path can be designed, then we can optimize each path, based on the current belief map, to maximize information gain about habitat existence.

The general concept of path-based sensors is introduced in [18], where it is proposed as a solution to the problem of gathering information about hazards in a communication-denied environment. Expendable agents are sent along predefined paths through the environment and their survival versus apparent destruction is used to gather information about hazards. A path-based sensor reading of "0" (hazard not detected) occurs each time an agent successfully completes its path. A path-based sensor reading of "1" (hazard detected) occurs each time an agent does *not* return, i.e., such that it is assumed to have been destroyed somewhere *en route*. In [18], an approximate Bayesian belief map update is presented, as well as a path-planning algorithm that leverages this update within a Shannon information-based framework. The related problem of target search in a hazardous communication-denied environment is also considered — where targets, e.g., search and rescue victims, are different from hazards.

Other work on path-based sensors has considered deploying multiple homogeneous agents [26] and heterogeneous agents [25] simultaneously, having a small finite number of agents [13], and cases in which deviation from the intended path is beneficial [14]. The prior body of work relies on an approximation to calculate the posterior belief map in the event that a path-based sensor is triggered. In particular, the posterior belief maps of mutually exclusive histories (of where the agent may have been destroyed) are calculated separately, and then combined base on relative likelihood as estimated by the prior belief map. Yet, a preliminary investigation that we documented in a technical report [23] indicated that better results are possible by carefully considering the correlations that exist between the alternative histories.

Contribution: We replace the approximation used in [18, 26, 13, 14, 25] with a Bayesian Network (BN) representation of a path-based sensor that considers a comprehensive joint probability space, accounting for static hazards, absence of hazards, agent survival, and non-survival along each path. We also show how the new update can be used within the iterative n-robot-at-a-time problem variant. We empirically compare the new method to prior work [18, 26] using simulations. Our simulations show that the new method improves the accuracy of the resulting Bayesian updates, accelerates convergence of the Bayesian belief map, and increases resilience to noisy data (false positives and false negatives) compared to [18] for single-robot and [26] for parallel deployment of multiple robots.

The rest of this paper is organized as follows. Related work is discussed in Section 2. The path-based sensor problem is formulated in Section 3 and the approaches to solving the problem are discussed in Section 4; the old approach used in prior work is outlined in Section 4.1, and the Bayesian Network approach is discussed in Section 4.2 (single robot) 4.3 (multi-robot). Algorithms are presented in Section 5. Simulation experiments and their results are discussed in Section 6 and conclusions are presented in Section 7.

2 Related Work

Bayesian frameworks have been extensively used in robotics for target estimation [5], where the posterior estimates of targets after observation of an event are more informative than the prior. Information theory [22] serves as the foundation for information-theoretic planners. The connection between Bayesian inference and information theory [17] has been foundational in the development of information-theoretic planners and inverse modeling problems [15]. Mutual information, a measure of relative information, has been used to improve estimation [2], for target tracking [7], and exploration [1].

With the increasing availability and reliability of autonomous systems, there has been a growing emphasis on utilizing multi-robot systems to tackle the problem of information gathering. Accurate and rapid estimation of hazard location was investigated initially in [21]. However, unlike the problem we consider, inter-agent communications are allowed in [21], which facilitates instant identification of false positives. While some prior work has considered multi-robot systems in environments where communication is strictly prohibited [6, 26], other work [3, 20, 30, 12, 9, 29] considers localizing hazards in environments with limited communication. For example, [10] provides a formal derivation of the gradient of mutual information, demonstrating that under a multi-agent gradient ascent control strategy, information entropy would approach zero at the limit. The latter formulations consider that the observations of interest are explicitly provided in real-time, while we account for an implicit type of path-based sensor observations that occur after each path has been traversed.

Probabilistic graphical models are used to represent the conditional dependencies among the involved random variables [19]. Bayesian networks have been used in numerous applications such as bio-informatics [31], machine learning classifiers [4], computer vision [16], and deep learning [8]. In the domain of environment estimation, a probabilistic graphical model-based method is used for tracking targets in [27] and [28]. We seek to detect hazards by estimating the origin of path-based sensor realizations using probabilistic graphical modeling.

Path-based sensors are theoretical sensors that provide binary observations to indicate the occurrence of an event along a traversed path, without reporting precise event locations [18]. These sensor observations are particularly relevant in scenarios where the occurrence of an event can only be determined in post-processing. For example, when wet-lab analysis is required to determine whether a chemical or biological agent was encountered, and when robots are deployed to detect static hazards in a communication denied environment [18, 26, 13, 14].

The current paper goes beyond prior work on path-based sensors by exploring a reformulation of the iterative map update procedure using tools from Bayesian networks. Some of the ideas in the current paper were presented in a non-archival workshop [24] and also documented in a technical report [23]. The current paper goes beyond [24, 23] by providing a more rigorous algorithmic formulation, expanded experiments, deeper discussion of the results, as well as showing how the new update can be used within the iterative n-robot-at-a-time problem variant.

3 Problem Formulation

We consider a search space $\mathrm{S} \subset \mathbb{R}^2$ composed of a $h \times w$ grid of discrete cells. Let C denote a single cell within the grid. Hazardous elements are present in S. We use Z to denote the state of the environment with respect to hazards. Z is a discrete time random variable that takes values on the alphabet $\mathcal{Z} = \{0,1\}^{h\times w}$; in other words, $\mathcal{Z}$ is an event space that contains each of the $2^{h\times w}$ different possibilities of having a hazard $\{1\}$ or not $\{0\}$ in each of the $h \times w$ separate cells of S. For tractability, we assume that hazard states are static and statistically independent across cells; modeling spatial or temporal dependencies would introduce additional edges in the Bayesian network and significantly increase the complexity of inference. Our belief $\mathbb{P}(Z)$ of hazard state is updated after each search round. When necessary for the sake of clarity, we use superscripts, e.g., $Z^{(i)}$, to indicate the belief map that incorporates information from search rounds 1 through i. Our prior belief of hazard state prior to any search round is denoted $Z^{(0)}$. Hazard state in cell C is denoted Z_C and hazard state in the subspace $\hat{\mathrm{S}} \subset \mathrm{S}$ is denoted $Z_{\hat{\mathrm{S}}}$. Similarly, our beliefs regarding these discrete time random variables are denoted $\mathbb{P}(Z_C = z_C)$ and $\mathbb{P}(Z_{\hat{\mathrm{S}}} = z_{\hat{\mathrm{S}}})$, respectively.

We assume that hazards can only be detected indirectly when agents fail to return after a search round, and that the destruction of agents cannot be observed directly. Even though hazard presence cannot be observed directly, it is convenient to let $Z_C = 1$ and $Z_C = 0$ denote the existence or absence of a hazard in cell C, respectively. We use Δ_C to denote the random variable of destruction in a given cell C, i.e. $\Delta_C = 1$ denotes that the agent was destroyed in cell C whereas $\Delta_C = 0$ denotes that the agent survived a passage through cell C. We use the variable $\delta_C \in \{0,1\}$ to denote the realization of such an event.

Agents are deployed to gather information about hazards by attempting to traverse paths through the environment. The environment contains a set of m base stations $D = \{d_1, d_2, \ldots, d_m\} \subset \mathrm{S}$. Agents start and end their search rounds at their designated base stations and cannot communicate from the field *during* a path traversal. However, reliable communication is assumed to exist among base stations and with a central server σ. Thus, knowledge about whether an agent i survives its path ζ_i is stored at σ and available at all base stations.

Definition 1. *Path-based sensor: A binary sensor that reports the occurrence of an event along a path but does not provide the location (i.e., cell) of occurrence.*

The random variable associated with an agent i surviving a path ζ_i is denoted Θ_i. The destruction or survival of agent i along path ζ_i is a path-based sensor (Definition 1) observation $\theta_i \in \{1,0\}$, where $\theta_i = 0$ is survival (hazard not observed) and $\theta_i = 1$ is destruction (hazard observed). Belief of path survival is thus $\mathbb{P}(\Theta_i = \theta_i)$. We consider cases where the path-based sensor may report false positives or false negatives. A false-positive accounts for a faulty agent that fails to traverse a path regardless of the presence of hazards, whereas a false-negative accounts for the chance of an agent surviving a hazardous cell.

The Shannon information entropy of the belief of hazard existence in cell C is defined as $H(Z_C) \coloneqq -\sum_{z_C \in \{0,1\}} \mathbb{P}(Z_C = z_C) \log \mathbb{P}(Z_C = z_C)$. Similarly, $H(Z_{\hat{\mathrm{S}}})$ is the Shannon information entropy of the entire subspace $\hat{\mathrm{S}}$. As in [18], we assume that hazard effects are local to each cell. This enables us to calculate $H(Z_{\hat{\mathrm{S}}})$ in $\mathcal{O}(h \times w)$ time by summing over the entropy of cells $C \in \hat{\mathrm{S}}$, instead of the $\mathcal{O}(2^{h\times w})$ time required, in general, to consider all $Z \in \mathcal{Z}$.

$$H(Z_{\hat{\mathrm{S}}}) = \sum_{C\subset\hat{\mathrm{S}}} H(Z_C) = -\sum_{C\subset\hat{\mathrm{S}}}\sum_{z_C\in\{0,1\}} \mathbb{P}(Z_C = z_C)\log\mathbb{P}(Z_C = z_C).$$

Observe that a path ζ traverses cells in the order $C_1,\ldots,C_\ell$ and revisiting cells is allowed such that C_k and C_j may or may not be the same. For ease of presentation, we abuse the notation and allow a path to be interpreted as the sequence of cells through which it traverses $\zeta = \langle C_1,\ldots,C_\ell\rangle$. The expected information gain about the presence of hazards can only be obtained indirectly through path-based sensor observations indicating agent destruction. Given a sensor observation Θ_i, the information gain is defined as the expected reduction in the entropy associated with Z,

$$I(Z^{(i)};\Theta_i) = H(Z^{(i-1)}) - H(Z^{(i)}\,|\,\Theta_i, Z^{(i-1)}),$$

where $H(Z^{(i)}\,|\,\Theta_i, Z^{(i-1)}) = \sum_{\theta_i\in\{0,1\}}\mathbb{P}(\Theta_i=\theta_i)H(Z^{(i)}\,|\,\Theta_i=\theta_i, Z^{(i-1)})$ is the conditional entropy computed by summing over the possible outcomes of $\Theta_i=\theta_i$ weighted by our current belief of their probability of occurring.

3.1 One Agent per Iteration Problem Formulation

In the classical version of the problem [18], agents are deployed iteratively, one-at-a-time, to gather information from the search space S. In the one-at-a-time scenario, we use the index i to indicate the agent that is deployed at iteration i. Agent i has knowledge of the survival status of all agents $1,\ldots,i-1$ and up-to-date access to the corresponding hazard belief $\mathbb{P}(Z^{(i-1)})$ that reflects this knowledge. We now formalize the iterative version of the problem.

Problem 1. Iterative path-based sensing for hazard detection in communication-denied environments: Given a search space S in which communication is prohibited from the field and where S contains an unknown number of hazards and base stations m, hazards can only be detected indirectly through the act of destroying agents, and assuming that the base stations can communicate with a central server σ, then iteratively deploy agents along paths $\zeta_1,\ldots,\zeta_{i-1}$ and observe whether or not the agents survive to gather information about hazards to refine hazard belief $\mathbb{P}(Z)$.

Assuming that we can choose the paths that agents traverse, the problem of path optimization can also be formalized.

Problem 2. Iterative information-optimal path-based sensor shape design: Given a prior belief $\mathbb{P}(Z)$ calculated from path-based sensor readings associated with $\zeta_1,\ldots,\zeta_{i-1}$, calculate a new path $\zeta^\star$ for agent i of length ℓ_i that maximizes the expected information gained from the path-based sensor Θ that will be generated by agent i following path ζ_i,

$$\zeta^\star = \arg\max_{\zeta_i} I(Z;\Theta_i). \quad (1)$$

3.2 n-Agents per Iteration Problem Formulation

In [26], n separate agents are deployed simultaneously during each iteration; however, agents are deployed in non-overlapping regions for the sake of parallel actuation as well as the convenience of the individual path calculations. In this multi-agent per iteration case, there is no longer a one-to-one mapping between iteration index and agent index. So, in the multi-agent case, we will refer to such iterations as *rounds* and use r to denote the current round. While i is still used to track agents and their paths, it now refers to the i-th agent that is deployed in a particular search round r. During each round r, the map is partitioned into n regions of approximately equal entropy corresponding to each agent. Each region contains at least one base station. The map partitioning changes as r increases due to changing information value on different cells.

During round r, each agent i starts at base station $d_i^{(r)}$ and is assigned the corresponding partition (subregion) as its search space $\mathrm{S}_i^{(r)} \subset \mathrm{S}$. In each round r, the union of all partitions forms the entire search space $\bigcup_{i=1}^{n} \mathrm{S}_i^{(r)} = \mathrm{S}$, and there is no overlap between partitions, $\mathrm{S}_i^{(r)} \cap \mathrm{S}_j^{(r)} = \emptyset$ for $i \neq j$. An agent is not expected to search the entire space in a single round, rather, it considers only cells within its search space partition when calculating its path $\zeta_i^{(r)} \subseteq \mathrm{S}_i^{(r)}$.

Each agent i is capable of visiting ℓ_i cells within its assigned search space, forming a path $\zeta_i^{(r)} = \langle C_1, C_2, \ldots, C_{\ell_i} \rangle \subseteq \mathrm{S}_i^{(r)}$. Note that in our experiments, we consider the case where all agents have an identical path length constraint $\ell_i = \ell$. We assume that there is no communication between the centralized server σ and the agents about their whereabouts until the search round r concludes.

Problem 3. Simultaneous iterative path-based sensing for hazard detection in communication-denied environments with simultaneous deployments: Given a search space S in which communication is prohibited from the field that contains an unknown number of hazards and where S has been divided into n non-overlapping regions $\mathrm{S}_1, \ldots, \mathrm{S}_n$, such that each region S_i contains at least one base station, and assuming that the base stations in all regions can communicate with a central server σ, then iteratively deploy agents n-at-a-time along paths $\zeta_1^{(r)}, \ldots, \zeta_n^{(r)}$ for rounds $r = 1, 2, \ldots$ such that $\zeta_i^{(r)} \subseteq \mathrm{S}_i^{(r)}$ for each r (and all i), and observe whether or not the agents survive to gather information about hazards to refine a hazard belief map Z.

When we consider Problem 3 in our experiments (Section 6), we assume that exploration always begins and ends at each agent's respective base station d_i and there is only one base station per partition ($m = n$). However, these requirements can be relaxed, in general.

4 Path-Based Sensors with Bayesian Networks

In this section we present a methodology based on Bayesian networks for estimating hazard locations in the search space. We overview the hazard belief map update mechanism with path-based sensors [18]. Next, we propose a Bayesian network modeling method for sequential deployment of multiple robots to improve the information gathering rate. Subsequently, we discuss the integration of the Bayesian network method within the parallel multi-agent information gathering problem to improve the computational scalability.

4.1 Weighted-Average Path-Based Sensors

Upon observing a path-based sensor trigger ($\Theta_i = 1$), the method in [18] computes the posterior belief map using a weighted-average technique. The method calculates the posterior belief maps produced by the mutually exclusive occurrence of the agent being destroyed at each cell along the path ζ_i. The final belief map (as calculated in [18]) is then found by combining the computed belief maps weighted by their relative likelihood of occurrence. Thus, the belief map $\mathbb{P}(Z^{(i)})$ at search round i is given by,

$$\mathbb{P}(Z^{(i)}) = \sum_{k=1}^{\ell_i} \mathbb{P}(Z^{(i)} | Z^{(i-1)}, \Delta_{\zeta_i,k} = 1) \mathbb{P}(\Delta_{\zeta_i,k} = 1), \tag{2}$$

where $\Delta_{\zeta_i,k} = 1$ denotes the event of path-based sensor triggering at the k-th cell of path ζ_i.

4.2 Bayesian Network Path-Based Sensors

We now propose a different methodology that uses a Bayesian network to incorporate the multi-universe structure into a single joint-distribution. Our methodology is comprised of three steps: (i) observation; (ii) inference; and (iii) estimation. Based on the prior belief map of the environment, our method *estimates* the new belief map after *inferring* the origin of the *observation* (i.e., location of agent destruction).

Consider that an agent traverses a path $\zeta_i = \langle C_1, C_2, \dots C_\ell \rangle$ and never reaches its designated goal cell C_ℓ. The problem can be modeled in a Bayesian network (Fig. 2-(a)). The upper layer containing $\boldsymbol{Z}_\zeta = Z_1, Z_2, \dots, Z_\ell$ is referred to as the *estimation* layer, where $\boldsymbol{Z}_\zeta$ denotes the random variables of hazard existence at cells $C_1, C_2, \dots, C_\ell$ along the path ζ_i. The variables in the estimation layer are each connected to $\boldsymbol{\Delta}_\zeta = \Delta_1, \Delta_2, \dots, \Delta_\ell$, where $\boldsymbol{\Delta}_\zeta$ denotes the random variables of destruction in cells $C_1, C_2, \dots, C_\ell$. This layer is referred to as the *inference* layer. The final layer is called the *observation* layer. Every element in the inference layer is connected to a single variable Θ (i.e., observation) which represents the random variable for triggering a path-based sensor.

This *simply-connected directed acyclic graph* (Fig 2-(a)) is a Bayesian network with joint distribution equal to the sum of the product for all possible distributions. For brevity we omit subscript i for the remainder of Section 4.1.

$$\mathbb{P}(Z, \Delta, \Theta) = \sum_{\substack{\boldsymbol{z} \in \{0,1\}^{h \times w} \\ \boldsymbol{\delta} \in \{0,1\}^{\ell} \\ \theta \in \{0,1\}}} \prod_{\substack{j \in [1, h \times w] \\ k \in [1, \ell]}} \mathbb{P}(Z_j = z_j, \Delta_k = \delta_k, \Theta = \theta) \tag{3}$$

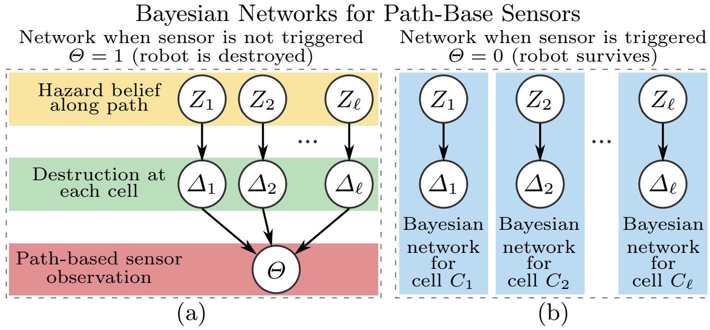


Fig. 2. Bayesian networks used for inference based on path-based sensor triggering. (a) Bayesian network used when the path-based sensor is triggered (the robot is destroyed, $\Theta = 1$) along a path of length ℓ. The *observation* layer (red) indicates whether the path-based sensor was triggered. The *inference* layer (green) speculates the most likely triggering cell based on the prior belief map. The *estimation* layer (yellow) represents the belief of hazards $\mathbb{P}(Z)$. The causal flow of this Bayesian network starts with the prior belief in the estimation layer, causing agent's destruction in the inference layer, triggering the path-based sensor observation in the *observation* layer. (b) The ℓ Bayesian networks used when the path-based sensor is not triggered (when the robot survives, $\Theta = 0$, it survives each of the ℓ cells along the path).

Given our assumption that hazard effects are local to each cell, (3) can be simplified by only considering the sequence of ℓ cells through which the path traverses, and not all $h \times w$ cells in the entire search space. The cell joint distribution is simplified to,

$$\mathbb{P}(Z_k, \Delta_k, \Theta) = \mathbb{P}(\Theta = \theta | \boldsymbol{\Delta}_\zeta = \boldsymbol{\delta})\mathbb{P}(\Delta_k = \delta_k | Z_k = z_k)\mathbb{P}(Z_k = z_k),$$

where $\mathbb{P}(\Theta = \theta | \boldsymbol{\Delta}_\zeta = \boldsymbol{\delta})$ is the likelihood of a path-based sensor trigger $\Theta = \theta$ along the path ζ, $\mathbb{P}(\Delta_k = \delta | Z_k = z)$ is the likelihood of destruction Δ_k given the existence of hazard $Z_k = z_k$, and $\mathbb{P}(Z_k = z_k)$ is the prior belief of the hazard Z_k at cell C_k of path ζ. The likelihood $\mathbb{P}(\Delta_k | Z_k)$ and the prior $\mathbb{P}(Z_k)$ have previously been used in various Bayesian Filters for updating belief maps and are derived from the application of Bayes' rule in similar environments. The likelihood term $\mathbb{P}(\Theta | \boldsymbol{\Delta}_\zeta)$ incorporates the *inference layer* of the Bayesian network. This term estimates the likelihood of tripping the path-based sensor given a specific permutation of ℓ plausible causes. The computation of this likelihood function is different for each specific application of path-based sensors.

In our case—where we assume the destruction of the agent as a path-based sensor triggering—the likelihood function results in,

$$\mathbb{P}(\Theta = 1 | \boldsymbol{\Delta} = \boldsymbol{\delta}) = \begin{cases} \prod_{k=1}^{j} \mathbb{P}(\Delta_k = \delta_k), & \text{if } \sum_{k=1}^{\ell} \delta_k \leq 1 \\ 0, & \text{if } \sum_{k=1}^{\ell} \delta_k > 1, \end{cases} \tag{4}$$

where j is the path index of the cell C_j at which $\Delta_j = 1$. Therefore, the likelihood function specifies the probability that the agent will be destroyed at C_j of ζ_i while surviving the $j - 1$ cells preceding it.

Remark 1. The conditions in (4) do not include repeated cell visiting along the path ζ_i. In other words, the path sequence $\zeta_i = \langle C_1, \ldots, C_\ell \rangle$ is the same as the ordered set $\{C_1, \ldots, C_\ell\}$. However, we allow the agent to have the flexibility to revisit cells during a path traversal. This alters the likelihood of destruction in a particular cell, as the agent may get destroyed at any possible instance of being located in that cell. Therefore, the new likelihood $\mathbb{P}(\Theta|\boldsymbol{\Delta})$ is calculated by summing the probability of each instance. Note that the Bayesian network shown in Fig. 2-(a) assumes that each variable is independent and corresponds to a specific cell of the path.

Given an observation of path-based sensor $\Theta = \theta$, the posterior of C_k cell of path ζ_i follows Bayes' rule,

$$\mathbb{P}(Z_k = 1|\Theta = \theta) \propto \sum_{\boldsymbol{z},\boldsymbol{\delta}\in\{0,1\}^\ell} \mathbb{P}(Z_k = 1, \Delta_k = \delta, \Theta) \prod_{j\neq k} \mathbb{P}(Z_j = z_j, \Delta_j = \delta, \Theta). \tag{5}$$

This unnormalized probability is normalized in the standard Bayesian manner. The calculation of the posterior probability of cell Z_k given the path-based sensor observation Θ using (5) is computationally expensive and becomes intractable as the path-length ℓ increases. The exact inference technique of variable elimination [11, Chapter 9] is used to address this problem. This technique eliminates all the random variable of hazards Z_j, where $j \neq k$, reducing the posterior probability of Z_k (5) to,

$$\mathbb{P}(Z_k = 1|\Theta = \theta) \propto \sum_{\boldsymbol{\delta}\in\{0,1\}} \mathbb{P}(\Delta_k = \delta|Z_k = 1)\mathbb{P}(Z_k = 1)\mathbb{P}(\Theta|\boldsymbol{\Delta} = \boldsymbol{\delta}) \prod_{j\neq k} \mathbb{P}(\Delta_j = \delta).$$

Note that (5) and (6) are relevant only when a path-based sensor is triggered, i.e. $\Theta = 1$. When $\Theta = 0$ the inference layer becomes redundant because each variable in the inference layer has a value of zero with probability equals 1, i.e. $\Delta_k = 0 \ \forall k \in \{1, \ldots, \ell\}$. This certainty that $\Delta = 0$ allows the Bayesian network to be broken into ℓ separate Bayesian networks, as shown in Fig. 2-(b). Thus, the joint distribution of Z_k and Δ_k is simplified as

$$\mathbb{P}(Z_k = 1, \Delta_k = 0) = \mathbb{P}(\Delta_k = 0|Z_k = 1)\mathbb{P}(Z_k = 1). \tag{6}$$

In addition, the posterior of k-th cell of path ζ_i in the event of no path-based sensor is triggered takes the form of

$$\mathbb{P}(Z_k = 1|\Delta_k = 0) = \frac{\mathbb{P}(\Delta_k = 0|Z_k = 1)\mathbb{P}(Z_k = 1)}{\mathbb{P}(\Delta_k = 0)}. \tag{7}$$

4.3 Multi-Robot Bayesian Network Path-Based Sensors

The Bayesian network path-based sensor formulation (presented in Section 4.2) can also be combined with the Distributed Entropy-weighted Voronoi Partition and Planner (DEVPP) framework described in [26]. A distributed strategy utilizes multiple agents in parallel in non-overlapping regions of the search space.

This is achieved in two steps: (i) entropy-based map partitioning; and (ii) local information-theoretic planning.

In (i), the search space S is partitioned into n regions, one for each robot to be deployed during the next search round r. The partitioning is accomplished by employing a weighted Voronoi partition mechanism with n base stations $d_1, \ldots, d_n$, serving as generator points. Let x_{d_i} denote the position of d_i and let x_C denote the position of the center of cell C. The entropy-weighted distance function used in [26] to help construct the entropy-weighted Voronoi partitions yields,

$$f(x_C, x_{d_i}) = \omega(Z_C, Z_{\mathrm{S}_i^{(r)}}) \ \ ||(x_C - x_{d_i})||_1,$$

where $||\cdot||_1$ is the L1 norm and $\omega(Z_C, Z_{\mathrm{S}_i^{(r)}})$ is the average entropy of the expected partition if cell $C \subset \mathrm{S}$ is added to subspace, and $\mathrm{S}_{i^{(r)}}$ is the weighted Voronoi region associated with base station d_i,

$$\omega(Z_C, Z_{\mathrm{S}_i^{(r)}}) = \frac{H(Z_C) + H(Z_{\mathrm{S}_i^{(r)}})}{\mathrm{card}(Z_{\mathrm{S}_i^{(r)}}) + 1}.$$

Thus, during round r the subspaces $\mathrm{S}_i^{(r)}$ for each agent i are constructed to contain approximately equal Shannon information entropy according to the rule,

$$\mathrm{S}_i^{(r)} = \bigcup_{C \subset \mathrm{S}} (C \mid f(x_C, x_{d_i}) \leq f(x_C, x_{d_j}), \forall\, i \neq j). \tag{8}$$

In step (ii), each agent i is assigned the task of finding an optimal path $\zeta_i^{(r)*}$ in its respective local search space $\mathrm{S}_i^{(r)}$. The objective is to maximize the expected information gain about hazards Z given the observed path-based sensor measurements Θ_{ζ_i} along path ζ_i at search round r. In contrast to the approach in [26], in the current paper each of these individual path planning problem is solved using Bayesian network modeling that is described in Section 4.2.

The two-step methodology is presented in Fig. 1, where (i) a central entity decomposes the environment into search spaces using an entropy-weighted Voronoi partitioning (Fig. 1-(a)); and (ii) each agent computes an informative path in its assigned region in a distributed manner to maximize information about hazards in their assigned search space using the proposed Bayesian network-based planner (Fig. 1-(b)). After each round, the centralized server updates the belief of the hazard state and assigns new search spaces with approximately equal information to each agent (Fig. 1-(c)). Then, the agents compute an informative path in their new search spaces to gather information about hazards Fig. 1-(d). Note that we also incorporate false-positive and false-negative sensor observations, making the model resilient to noisy data.

Following the computation of local informative paths, each agent traverses the planned paths. Then, the base stations observe the survival of each agent (i.e., path-based sensor) and communicate the observations to a central server that updates the belief map of the environment. This process is repeated at the end of each search round, where the methodology is applied recursively to compute new local search spaces based on the posterior belief map and make new path-based sensor observations as illustrated in Fig. 1.

5 Algorithms

In this section, we present algorithms that use our Bayesian network modeling approach to address Problems 1 and 3 by estimating the posterior hazard belief map of an unknown environment recursively. We call our algorithm that addresses Problem 1 "Bayesian Network-based Information Theoretic Planner" (BNITP), and our algorithm that addresses Problem 3 "Bayesian Network-based Distributed Entropy-Weighted Voronoi Partition and Planner" (BN-DEVPP). Because much of the logic of BNITP is also used in BN-DEVPP to solve Problem 3, it is convenient to break the presentation of BNITP into a high-level outer loop (Algorithm 1) that invokes a subroutine (Algorithm 2) once per iteration. While our presentation uses a maximum number of iterations i_{max}, it is possible to set this to ∞ and/or to use some other stopping criterion. The outer loop of BNITP simply calls the BNITP repeatedly.

The implementation of a single BNITP iteration is presented in Algorithm 2. We compute the relaxed optimal path ζ_i based on the prior hazard belief map $Z^{(i-1)}$ using the subroutine `calculateBNPath` (line 2). `calculateBNPath` calculates the path as originally described in [18], except that path-based sensor updates are performed according to our new Baysian networked-based method method. We present `calculateBNPath` in the appendix as Algorithm 4. Next, we observe the realization of the path-based sensor, where $\theta_i = 1$ if the agent is destroyed and $\theta_i = 0$ if the agent survives from the traversed path (line 3). When the path-based sensor is triggered ($\theta_i = 1$), the posterior hazard belief map $Z^{(i)}$ is updated using (5). Similarly, if the path-based sensor is not triggered ($\theta_i = 0$), then the posterior hazard belief map $Z^{(i)}$ is updated using (7). The use of (5) on line 4 is a key difference versus [18].

Algorithm 1 : One Agent per Iteration (BNITP-LOOP)

Inputs: hazard belief map prior $\mathbb{P}(Z^{(0)})$, search space S, base station set D, maximum number of iterations i_{max}
Output: $\mathbb{P}(Z^{(i_{max})})$

1: **for** $i = 1$ to i_{max} **do**
2: $\quad \mathbb{P}(Z^{(i)}) \leftarrow$ BNITP-iteration ($\mathbb{P}(Z^{(i-1)})$, S, D) // Algorithm 2

Algorithm 2 : One Agent per Iteration (BNITP-iteration)

Inputs: prior hazard belief map $\mathbb{P}(Z^{(i-1)})$, search space S, base station set D
Output: posterior hazard belief map $\mathbb{P}(Z^{(i)})$

1: $\zeta_i \leftarrow$ `calculateBNPath`$(Z^{(i-1)}, \mathrm{S}, D)$ // Algorithm 4, contains difference vs. [18]
2: observe θ_i after agent traverses path ζ_i
3: **if** $\theta_i = 1$ **then**
4: $\quad Z^{(i)} =$ `BN_PBSTrig`$(Z^{(i-1)}, \zeta_i, \theta_i)$ // (5), key difference vs. [18]
5: **else**
6: $\quad Z^{(i)} =$ `BN_NoTrig`$(Z^{(i-1)}, \zeta_i, \theta_i)$ // (7)

Algorithm 3 : n-Agents per Iteration BN-DEVPP

Inputs: hazard belief map prior $Z^{(0)}$, search space S, number of regions n, base station set D, maximum number of rounds $r_{\max}$
Output: $Z^{(r_{\max})}$

for $r = 1$ to $r_{\max}$ **do**
  calculate $\mathrm{S}_1^{(r)}, \ldots, \mathrm{S}_n^{(r)}$ // on server σ, (8)
  $Z^{(r-1)}_{\mathrm{S}_1^{(r)}} \cup \ldots \cup Z^{(r-1)}_{\mathrm{S}_n^{(r)}} \leftarrow Z^{(r-1)}$ // on server σ
  broadcast $\mathrm{S}_i^{(r)}$ and $Z^{(r-1)}_{\mathrm{S}_i^{(r)}}$ from σ to d_i // on server σ
  for $i = 1, \ldots, n$ **do** // in parallel
    $Z^{(r)}_{\mathrm{S}_i^{(r)}} \leftarrow$ BNITP-ITERATION$(Z^{(r-1)}_{\mathrm{S}_i^{(r)}}, D, \mathrm{S}_i^{(r)})$ // difference vs. [26]
    communicate $Z^{(r)}_{\mathrm{S}_i^{(r)}}$ from d_i to σ
  $Z^{(r)} \leftarrow Z^{(r)}_{\mathrm{S}_1^{(r)}} \cup \ldots \cup Z^{(r)}_{\mathrm{S}_n^{(r)}}$ // on server σ

Algorithm 3 outlines the details of the BN-DEVPP method. After each search round r, the centralized server σ initiates the algorithm by transmitting the prior belief map to all base stations $d_1, \ldots, d_n$. Subsequently, the central server computes the local search spaces S_i for each agent i, which are then communicated to their respective base stations. Each agent i employs the BNITP routine (Algorithm 2) to independently compute an information path that maximizes information gain within their local search space.

6 Experiments and Results

In this section, we report results from numerical experiments to evaluate the proposed methodologies. We compare the proposed techniques with [18] for sequential and with [26] for parallel deployment of multi-robot systems.

Experimental Setup: We consider a discrete spatial environment of dimension $h \times w = 15 \times 15$ cells that contain 7 hazards in unknown locations. Our task is to detect the location of the hazards by deploying multiple robots, either sequentially and/or in parallel. A cell in the environment C is either empty ($Z_C = 0$) or contains a hazard ($Z_C = 1$) that threatens to destroy the agent, i.e., trigger the path-based sensor. The agent's movement in the environment is determined by a 9-grid connectivity. This means that each agent can select the next step either by transitioning to any of its 8-neighboring cells or by remaining in the same cell. We consider two environments with *hazard lethality* of 70% and 90%. Hazard lethality refers to the likelihood of an agent's destruction if it reaches a cell C that is occupied by a hazard $Z_C = 1$. Each agent has a malfunction probability of 5% (false-positive) and 5% chances of surviving a cell C with a hazard (false-negative). We conduct Monte Carlo experiments in simulation to test the efficacy and robustness of the proposed methods, 15 trials are performed for each combination of method and number of agents deployed in parallel.

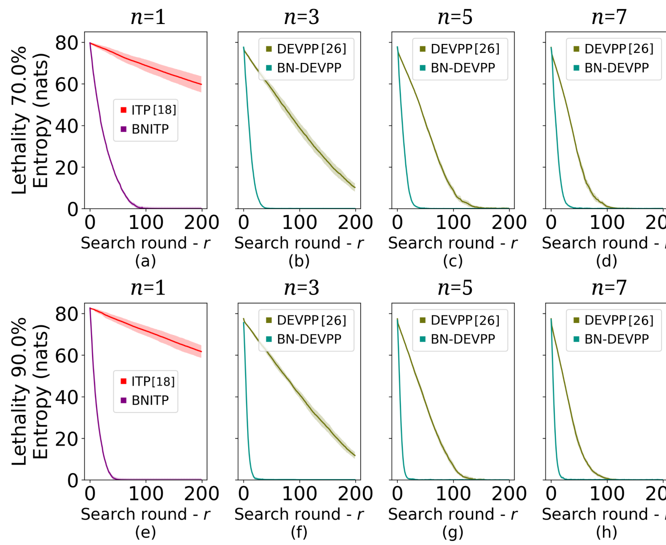


Fig. 3. Comparison of information entropy across search rounds for different methods: (a) BNITP vs. ITP [18] for 70% lethality; (b)-(d) BN-DEVPP vs. DEVPP [26] for 70% lethality and $n = 3, 5, 7$ respectively; (e) BNITP vs. ITP [18]; and (f)-(h) BN-DEVPP vs. DEVPP [26] for 90% hazard lethality and $n = 3, 5, 7$ respectively.

Table 1. Average Number of Agents Destroyed, for Various Methods and n

Lethality Rate 70%

n	DEVPP [26]	BN-DEVPP
1	$>$ 1000 [18]	**60.1**
3	464.9	**54.7**
5	294.1	**71.7**
7	261.2	**86.6**

Lethality Rate 90%

n	DEVPP [26]	BN-DEVPP
1	$>$ 1000 [18]	**25.2**
3	478.3	**26.6**
5	287.9	**40.3**
7	245.6	**51.2**

Results and Discussion: We conduct three experiments to study the efficacy of the proposed methodologies. In the first experiment, we compare the proposed methodologies with [18] and [26]. Fig. 3 illustrates the comparison of information entropy (i.e., negative uncertainty) of the environment in subsequent search rounds for all methods. For the case of sequential agent deployment ($n = 1$), we compare our BNITP algorithm with the ITP algorithm [18], while for parallel deployment with multiple agents ($n > 1$), we compare our BN-DEVPP with DEVPP [26]. The results demonstrate that as the number of agents increases, the information entropy reduces at a faster rate in both environments with 70% and 90% lethality. This means that the proposed methodologies (BNITP and BN-DEVPP) gather significantly more information about hazard locations in the first $r = 50$ search rounds. These findings suggest that the Bayesian network-based approach is more efficient at detecting hazards in communication-denied environments.

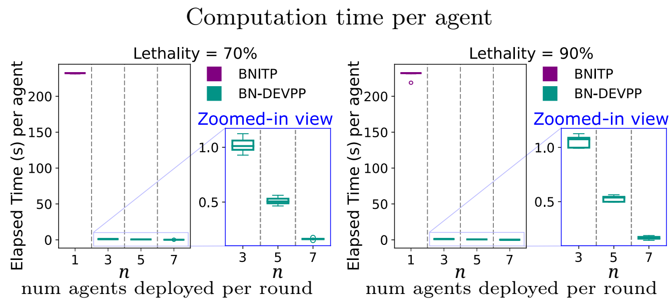


Fig. 4. Computation time per agent for different n. By definition $n = 1$ for BNITP and $n > 1$ for BN-DEVPP. Results for 70% (Left) and 90% (Right) lethal hazards. Box-plots show statistics over search rounds.

In the second experiment, we observe the number of agents lost in the hazardous environment until the information entropy decreased to 10% of its initial value at search round $r = 0$. Table 1 presents the average number of agents lost during the experiment. Our proposed methodology exhibits a notable reduction in the number of agents lost compared to previous methods. In the environment with 70% hazard lethality, we observe a 79.12% average reduction in the number of agents lost (averaged over $n = 3, 5, 7$), while in the environment with 90% lethality, we observe a reduction of 88.32% on average in agent destruction.

The third experiment investigates the time required per agent to compute a path for a search round. The findings are illustrated in boxplots in Fig. 4. We observe that increasing the number of agents leads to a reduction in the computation time for path planning. This scalability in computation time makes the algorithm suitable for real-time implementation. The results reveal the ability of the proposed method to accelerate the information gathering process, minimize agent losses, and enable scalable multi-agent deployment.

7 Conclusion

This paper presents a new Bayesian network modeling approach for path-based sensor updates. We introduce two algorithms, BNITP and BN-DEVPP, and through a series of experiments considering hazard detection in communication-denied environments, we compare the proposed method to prior work. We find that the proposed Bayesian network approach outperforms existing methods by achieving faster and more efficient information gathering while reducing the number of agents lost. In particular, the new method outperforms the sequential method from [18] in the sequential case, and n-robot-at-a-time method from [26] in the parallel case. We verify that a result from [26], that deployment of multiple agents in parallel improves scalability of path-based sensors algorithms, also holds for the new method. By distributing the information gathering tasks among multiple agents, the system benefits from parallelization and leads to improved efficiency and scalability, as the agents can cover larger areas without task duplicity. Overall, the Bayesian network formulation provides an efficient and robust framework for path-based sensors.

8 Acknowledgments

This work is supported by the Maryland Robotics Center and the Office of Naval Research (ONR) via grant N0001420WX01827 and N00014-20-1-2712. The views, positions, and conclusions contained in this document are solely those of the authors and do not explicitly represent those of ONR.

Appendix

The indices i, j, r used in the appendix match the notation used in [18] and they have different meanings than elsewhere in the current paper.

Algorithm 4 presents pseudocode for `calculateBNPath` which is a modified version of Algorithm 2 appearing in [18]. It performs a reverse search through a graph $(V_{\mathbb{S}\times\mathbb{T}}, E_{\mathbb{S}\times\mathbb{T}})$ describing the connectivity of S across time $\mathbb{T}$. A particular node $\nu_i \in V_{\mathbb{S}\times\mathbb{T}}$ represents visiting cell C at a particular time t. The edge $(\nu_i, \nu_j) \in E_{\mathbb{S}\times\mathbb{T}}$ connects node ν_i to ν_j (necessarily moving forward through time). The notation $\zeta_r \leftarrow (\nu_i, \nu_j) + \zeta_{\nu_j}$ indicates that the subpath ζ_r is constructed by prepending edge (ν_i, ν_j) to the front of subpath ζ_{ν_j}, where ζ_{ν_j} starts at node ν_j and ends at the node associated with the goal cell C_{goal}. The variable h is used to indicate the running expected information gain estimate, e.g., h_{ν_i} is the most expected information gained that has been found (so far) from any subpath starting from ν_i and ending at the goal. As described in [18], finding the true optimal path requires looking at all possible histories of reaching each node, which becomes intractable, even for small S. As in [18], tractability is achieved by a relaxation that iteratively considers only the set of subpaths that can be created by connecting nodes at time slice t to the best subpaths found starting at time slice $t+1$. We are also able to use a first-in-first-out queue because paths must move forward through time.

Algorithm 4 `calculateBNPath`

Inputs: prior hazard belief map $\mathbb{P}(Z^{(i-1)})$, search space S, nodes $V_{\mathbb{S}\times\mathbb{T}}$
Output: path ζ_r

for all $\nu_i \in V_{\mathbb{S}\times\mathbb{T}}$ **do**
  $h_{\nu_i} \to -\infty$
`InsertFIFOQueue`(C_{goal})
while $\nu_i \leftarrow$ `PopFIFOQueue` **do**
  for all $(\nu_i, \nu_j) \in E_{\mathbb{S}\times\mathbb{T}}$ **do**
    $\zeta_r \leftarrow (\nu_i, \nu_j) + \zeta_{\nu_j}$
    // Equation (5), key diff. vs. [18]:
    $(\mathbb{P}(Z_{\text{live}}), p^{\text{alive}}_{\zeta_r}) \leftarrow$
      `BN_PBSTrigger`$(\mathbb{P}(Z^{(r-1)}), \zeta_r, 1)$
    // Equation (7):
    $\mathbb{P}(Z_{\text{killed}}) \leftarrow$
      `BN_NoTrigger`$(Z^{(r-1)}, \zeta_r, 0)$
    $h_{this} \leftarrow p^{alive}_{\zeta} H(Z_{live})$
      $+(1 - p^{alive}_{\zeta}) H(Z_{killed})$
    if $h_{this} > h_{\nu_i}$ **then**
      $\zeta_{\nu_i} \leftarrow \zeta_r$
      $h_{\nu_i} \leftarrow h_{this}$
    for all $(\nu_k, \nu_i) \in E_{\mathbb{S}\times\mathbb{T}}$ **do**
      if not `InQueue`(ν_k) **then**
        `InsertFIFOQueue`(ν_k)
$w_{start} \leftarrow \arg\min_{w \in W_{start}} h_w$
$\zeta_r \leftarrow \zeta_{w_{start}}$
return ζ_r

References

1. Bourgault, F., Makarenko, A.A., Williams, S.B., Grocholsky, B., Durrant-Whyte, H.F.: Information based adaptive robotic exploration. In: IEEE/RSJ International Conference on Intelligent Robots and Systems. vol. 1, pp. 540–545 (2002)
2. Charrow, B., Kumar, V., Michael, N.: Approximate representations for multi-robot control policies that maximize mutual information. Autonomous Robots **37**(4), 383–400 (2014)
3. Flint, M., Fernandez-Gaucherand, E., Polycarpou, M.: Cooperative control for UAV's searching risky environments for targets. In: IEEE International Conference on Decision and Control. vol. 4, pp. 3567–3572 (2003)
4. Friedman, N., Geiger, D., Goldszmidt, M.: Bayesian network classifiers. Machine learning **29**(2), 131–163 (1997)
5. Furukawa, T., Bourgault, F., Lavis, B., Durrant-Whyte, H.: Recursive Bayesian search-and-tracking using coordinated UAVs for lost targets. In: IEEE International Conference on Robotics and Automation. pp. 2521–2526 (2006)
6. Gielis, J., Shankar, A., Prorok, A.: A critical review of communications in multi-robot systems. Current Robotics Reports pp. 1–13 (2022)
7. Grocholsky, B.: Information-theoretic control of multiple sensor platforms. Ph.D. thesis, University of Sydney. School of Aerospace, Mechanical and Mechatronic Engineering (2002)
8. Hao, W., Yeung, D.Y.: Towards Bayesian deep learning: A framework and some existing methods. IEEE Transactions on Knowledge and Data Engineering **28**(12), 3395–3408 (2016)
9. Jorgensen, S., Chen, R.H., Milam, M.B., Pavone, M.: The matroid team surviving orienteers problem: Constrained routing of heterogeneous teams with risky traversal. In: IEEE/RSJ International Conference on Intelligent Robots and Systems. pp. 5622–5629 (2017)
10. Julian, B.J., Angermann, M., Schwager, M., Rus, D.: Distributed robotic sensor networks: An information-theoretic approach. The International Journal of Robotics Research **31**(10), 1134–1154 (2012)
11. Koller, D., Friedman, N.: Probabilistic graphical models: Principles and techniques. MIT press (2009)
12. Lyu, Y.H., Chen, Y., Balkcom, D.: k-survivability: Diversity and survival of expendable robots. IEEE Robotics and Automation Letters **1**(2), 1164–1171 (2016)
13. McGuire, L., Otte, M., Sofge, D.: Valuing attrition in a fleet of robots used as path-based sensors for gathering information in a communications restricted environment. In: IEEE/RSJ International Conference on Intelligent Robots and Systems. pp. 7633–7640 (2024)
14. Mendelsohn, A., Sofge, D., Otte, M.: Enhancing search and rescue capabilities in hazardous communication-denied environments through path-based sensors with backtracking. In: International Conference on Autonomous Agents and Multiagent Systems. pp. 2387–2389 (2024), (extended abstract)
15. Mohammad-Djafari, A.: Entropy, information theory, information geometry and Bayesian inference in data, signal and image processing and inverse problems. Entropy **17**(6), 3989–4027 (2015)
16. Nie, S., Zheng, M., Ji, Q.: The deep regression Bayesian network and its applications: Probabilistic deep learning for computer vision. IEEE Signal Processing Magazine **35**(1), 101–111 (2018)

17. Oladyshkin, S., Nowak, W.: The connection between Bayesian inference and information theory for model selection, information gain and experimental design. Entropy **21**(11), 1081 (2019)
18. Otte, M., Sofge, D.: Path-based sensors: Paths as sensors, Bayesian updates, and Shannon information gathering. IEEE Transactions on Automation Science and Engineering **18**(3), 946–967 (2021)
19. Pearl, J.: Probabilistic reasoning in intelligent systems: Networks of plausible inference. Morgan kaufmann (1988)
20. Sato, H., Royset, J.O.: Path optimization for the resource-constrained searcher. Naval Research Logistics (NRL) **57**(5), 422–440 (2010)
21. Schwager, M., Dames, P., Rus, D., Kumar, V.: A multi-robot control policy for information gathering in the presence of unknown hazards. In: Robotics Research, pp. 455–472. Springer (2017)
22. Shannon, C.E.: A mathematical theory of communication. The Bell System Technical Journal **27**(3), 379–423 (1948)
23. Srivastava, A.K., Kontoudis, G.P., Sofge, D., Otte, M.: Path-based sensors: Will the knowledge of correlation in random variables accelerate information gathering? (2023), https://arxiv.org/abs/2305.06929
24. Srivastava, A.K., Kontoudis, G.P., Sofge, D., Otte, M.: Path-based sensors: Will the knowledge of correlation in random variables accelerate information gathering? In: IEEE International Conference on Robotics and Automation, Workshop on Communication Challenges in Multi-Robot Systems: Perception, Coordination, and Learning (2023)
25. Srivastava, A.K., Suresh, A., Nieto-Granda, C.: Behaviorally adaptive multi-robot hazard localization in failure-prone, communication-denied environments. arXiv preprint arXiv:2508.04537 (2025)
26. Srivastava, A.K., Kontoudis, G.P., Sofge, D., Otte, M.: Distributed multi-robot information gathering using path-based sensors in entropy-weighted Voronoi regions. In: International Symposium on Distributed Autonmous Robotic Systems (2022)
27. Uney, M., Cetin, M.: Graphical model-based approaches to target tracking in sensor networks: An overview of some recent work and challenges. In: International Symposium on Image and Signal Processing and Analysis. pp. 492–497 (2007)
28. Uney, M., Cetin, M.: Target localization in acoustic sensor networks using factor graphs. In: IEEE Signal Processing, Communication and Applications Conference. pp. 1–4 (2008)
29. Yang, Y., Minai, A.A., Polycarpou, M.M.: Decentralized cooperative search by networked UAVs in an uncertain environment. In: American Control Conference. vol. 6, pp. 5558–5563 (2004)
30. Yang, Y., Polycarpou, M.M., Minai, A.A.: Multi-UAV cooperative search using an opportunistic learning method. Journal of Dynamic Systems, Measurement, and Control **129**(5), 716–728 (2007)
31. Yu, J., Smith, V.A., Wang, P.P., Hartemink, A.J., Jarvis, E.D.: Advances to Bayesian network inference for generating causal networks from observational biological data. Bioinformatics **20**(18), 3594–3603 (2004)