Sherlock Bayes: The Curious Case of the Vanishing Posterior

Surajit Dutta

Email ralopezm@geologia.unam.mx
https://doi.org/10.61618/LLSH2304

Abstract

Search and rescue (SAR) planning repeatedly confronts the same operational challenge: how to allocate limited search effort across sectors when information is incomplete and detection is imperfect. This paper presents a Bayesian decision-support framework for SAR tasking that maintains a spatial belief map over a discretized search region (interpretable as Probability of Area, POA) and updates it after each unsuccessful search using Bayes' theorem, explicitly accounting for Probability of Detection (POD) that may vary by terrain, access, and sensing modality. At each step, the framework prioritizes the sector maximizing POA × POD, i.e., the sector with the highest immediate Probability of Success (POS). The title's "vanishing posterior" refers to a key operational insight: when POD < 1, a negative result does not drive a sector's POA to zero, formalizing why re-search decisions remain rational in low-detectability terrain. The paper further outlines practical extensions relevant to SAR: (i) terrain-aware POD models, (ii) informed priors derived from geospatial or historical cues, (iii) belief propagation for moving subjects, and (iv) a POMDP framing for longer-horizon planning under explicit costs. Proof-of-concept simulations illustrate how POA concentrates over time and how imperfect detection slows clearance of difficult terrain. Limitations of grid-based simulations are acknowledged, and a worked hypothetical SAR example demonstrates how the framework can support sector prioritization and allocation of effort in planning practice.

KEY WORDS: *Bayes' theorem; Search and rescue (SAR); Probability of Area (POA); Probability of Detection (POD); Probability of Success (POS); Imperfect detection; Terrain-aware search; Belief updating; Dynamic subject modelling*

1. Introduction

1.1. Why this problem matters in SAR tasking

SAR operations must make sequential tasking decisions under uncertainty: where to assign a team next, whether to re-search a sector, and how to distribute effort across terrain when subject location is unknown and detection is imperfect. In the field, a negative result rarely "clears" an area absolutely; instead, it provides evidence whose strength depends on how thoroughly the area was searched and how detectable the subject would have been under the conditions. This motivates a decision-support approach that (i) represents uncertainty explicitly, (ii) updates beliefs as evidence accumulates, and (iii) links those beliefs to a transparent tasking rule.

This manuscript develops such a framework by maintaining a probability distribution over sectors and updating it after each negative result using Bayes' theorem—capturing the operational reality that not found is informative but not definitive when POD is less than 1. The aim is not to replace SAR expertise, but to provide a principled and auditable way to aggregate evidence and justify prioritization as a mission evolves.

1.2. A SAR-native glossary: POA, POD, POS

To keep the analysis aligned with SAR planning language, we use the following mapping throughout:

• Sector / grid cell: a taskable search segment (the unit of assignment).

• POA (Probability of Area): the probability the subject is in sector i, written $b_t(i)$ at time t (1).

• POD (Probability of Detection): the probability of detecting the subject if the subject is in sector i and the sector is searched with a specified resource/technique, written p_i (1).

• POS (Probability of Success): the probability a search of sector i succeeds immediately: $POS_t(i) = b_t(i)\ p_i$. This is the immediate expected detection probability and a natural basis for prioritization (1).

This POA–POD–POS mapping allows the Bayesian belief map to be read directly as an operational prioritization surface and makes the method interpretable to a mixed practitioner–academic readership.

1.3. What is the "vanishing posterior," and why it matters operationally

A common misconception in practice is that an unsuccessful search implies a sector is "empty." Under imperfect detection, Bayesian updating formalizes the opposite: after searching sector j and not finding the subject, the posterior POA of j decreases, but does not vanish unless POD is perfect ($p_j = 1$). This provides a quantitative rationale for re-searching difficult sectors (dense forest, poor visibility, inaccessible terrain) where POD is low and a negative result should be weighted cautiously.

1.4. Paper objective and scope

Objective. The primary objective is to present a mathematically correct, SAR-interpretable Bayesian framework for sequential search tasking that updates POA using negative search outcomes and prioritizes future tasking using POS.

Scope. We demonstrate proof-of-concept behaviour using grid-based simulations and then discuss how the same framework can be strengthened for real-world deployment via terrain-aware POD calibration, informed priors from SAR intelligence inputs, moving-subject models, and longer-horizon planning formulations.

1.5. Contributions

This paper contributes:

Methodological contributions

1. A clear Bayesian belief-update rule for negative search outcomes under imperfect detection, with explicit assumptions and normalization.

2. A POS-based sector selection rule (maximize POA × POD) that is transparent and directly interpretable for tasking decisions.

SAR-facing contributions

3. An operational interpretation of belief updating—how POD and terrain influence sector prioritization, re-search decisions, and allocation of effort.

4. A worked hypothetical SAR example showing how the framework might be applied in practice.

5. A limitations and deployment pathway section specifying what additional data, calibration, and validation are required before operational use.

How this article differs from SAROPS / existing planning: (i) closed-form sector update, (ii) explicit re-search logic under POD<1, (iii) field-deployable spreadsheet/briefing workflow, (iv) minimal assumptions stated as a 'model contract'.

1.6. Roadmap

Section 2 positions the work within SAR-oriented Bayesian and operational search-planning literature. Section 3 formalizes the problem using POA/POD language. Section 4 derives the Bayesian update and the POS tasking rule. Section 5 presents the algorithm and flow logic. Sections 6–9 introduce terrain-aware POD, informed priors, moving-subject extensions, and a POMDP framing for longer-horizon planning. Section 10 provides proof-of-concept simulation results and visualization. Section 11 presents a worked hypothetical SAR example. Sections 12–13 discuss operational implications and limitations, respectively, before concluding.

2. Related Work (Operational / Practitioner-Focused)

Search planning in search and rescue (SAR) is built around a small set of operational probability concepts—where the subject is likely to be, how detectable they are under current conditions, and how to allocate scarce effort to maximize the chance of finding them quickly. U.S. Coast Guard guidance summarizes the scientific basis behind these ideas and emphasizes that improved planning methods primarily refine how uncertainty, detectability, and prior searches are integrated into decisions. This section synthesizes (i) the operational POA/POD/POS framework used by SAR planners, (ii) Bayesian updating as implemented in real decision-support tools (notably SAROPS), and (iii) land/WiSAR developments that highlight terrain-driven variability in detectability and the role of GIS-based priors (5,6).

2.1 Operational SAR search theory: POA/POC, POD, POS, and the meaning of a "negative result"

A core operational principle is that SAR success depends on searching areas that actually contain the subject and using search methods that can detect the subject given local conditions. The U.S. Coast Guard's Theory of Search formalizes this using probability maps and detection models, centering on (i) Probability of Area/Containment (POA/POC), (ii) Probability of Detection (POD), and (iii) Probability of Success (POS) as a practical measure of search effectiveness (1).

Foundational context. The general probabilistic foundations of search theory (including detection functions and the logic of allocating effort under uncertainty) originate in Koopman's classic synthesis, and have been further developed in standard treatments such as Stone's and Washburn's texts. (2–4)

Crucially, operational doctrine treats "not found" as evidence, not proof of absence: the impact of a negative result depends on detectability and coverage. Coast Guard guidance develops detection models (including sweep width and related constructs) to quantify how much confidence should be gained—or not gained—after an unsuccessful search, motivating why rigorous planning can improve allocation and sequencing (8,9).

2.2 Decision-support in practice: CASP → SAROPS and Bayesian incorporation of unsuccessful searches

Operational SAR planning has a strong tradition of computerized decision-support, especially in maritime contexts. The Search and Rescue Optimal Planning System (SAROPS) is the U.S. Coast Guard's operational tool for maritime search planning and is described as the successor to earlier Bayesian planning systems (CASP). SAROPS uses Monte Carlo/particle-based simulation to produce probability distributions (maps) of object location and incorporates unsuccessful searches in a Bayesian fashion to form posterior distributions that drive subsequent planning cycles (5,6).

A practitioner-relevant aspect of SAROPS is that detectability is not a single constant: SAROPS models probability of detection using sensor/platform-specific functions (e.g., lateral range curves), and it updates particle weights based on the probability that a Search and Rescue Unit would have failed to detect an object under environmental conditions. Coast Guard descriptions frame SAROPS as maximizing probability of success while accounting for drift, environment, and effects of previous searches (5,6).

The present manuscript can be seen as a deliberately simplified, transparent analogue of these operational cycles: maintain a POA-like belief surface, model POD per sector/terrain, update POA after unsuccessful search, and prioritize tasking using an immediate POS criterion.

2.3 Land SAR and WiSAR: calibrating POD in the field and avoiding "one-size-fits-all detectability"

In land and wilderness SAR (WiSAR), detectability varies dramatically with terrain, vegetation density, lighting, clue visibility, and searcher capability. Empirical work in the Journal of Search and Rescue demonstrates this explicitly: Koester reports sweep-width and detectability changes between daytime and night searching and shows how coverage–POD relationships can be validated against field experiments, underscoring that POD is strongly context-dependent and must be treated as a modeled quantity rather than assumed uniform (8,9).

Operational reviews sponsored by U.S. Coast Guard / DHS have also noted that land SAR procedures have not always incorporated formal search theory consistently and argue for standardized methodologies that connect probability maps, POD estimation, and effort allocation to operational decision-making (7).

2.4 Bayesian GIS and evidence fusion for SAR planning (worked-case orientation) (11)

A recurring practical problem is that initial probability maps often combine multiple weak or heterogeneous sources: last known position (LKP), witness reports, cellphone pings, trail networks, prior search tracks, terrain barriers, and subject-profile expectations. Practitioner-oriented case work in JSAR shows how heterogeneous evidence can be integrated into a probability search map using Bayes' theorem, directly supporting resource prioritization under uncertainty.

This is especially relevant for the present paper's emphasis on informed priors: operationally, priors are rarely uniform, and the quality of early tasking depends on how effectively planners encode and update these priors as new information arrives. SAROPS documentation similarly stresses scenario-based construction of priors and Bayesian updating after unsuccessful searches (5,6).

2.5 Decision-support systems and the practitioner–research interface (why clarity matters)

A systematic review of decision-support in SAR notes that SAR is a multi-actor, multi-agency process where GIS and analytics are frequently used (particularly in land rescue), and that decision-support tools aim to improve timeliness and quality of decisions under complexity. This reinforces a key requirement for JSAR readership: methods must remain interpretable and operationally grounded so probability outputs can be used in briefings, tasking meetings, and documentation of rationale (12).

2.6 Positioning of the present work

Unlike SAROPS' Monte Carlo/particle re-weighting, this sector-level belief map deliberately trades fidelity for transparency and auditability—POS = POA×POD can be read and briefed directly, and each update can be justified in tasking meetings and after-action reviews without a black-box step.

Relative to the practitioner-facing literature above, this paper positions its contribution as:

1. SAR-native belief updating under imperfect detection: a clear Bayesian update for negative search outcomes that matches POA/POD interpretation and explains why POA does not vanish when POD < 1.

2. Terrain-aware POD as an explicit planning lever: a sector-dependent detectability layer motivated by land SAR evidence enabling realistic interpretations of clearance and re-search decisions.

3. Informed priors and evidence fusion consistent with practice: a workflow that accommodates GIS-based cues and heterogeneous evidence, consistent with practitioner case studies and scenario-based maritime planning approaches.

4. A transparent tasking heuristic aligned with POS thinking: a prioritization rule based on immediate POS (POA×POD) with a clear path to richer optimization later.

SAROPS is a high-fidelity particle system; this paper provides a closed-form sector-level update + spreadsheet-ready workflow + explicit re-search logic under POD<1.

3. Problem Formulation (SAR tasking under imperfect detection)

This section formalizes the SAR planning cycle as a sector-tasking problem under uncertainty, using the operational concepts POA/POC, POD, and POS that underpin modern SAR search planning and decision support (1).

3.1 Search region, sectors, and state of the subject

We represent the search region as a set of taskable sectors (segments) indexed by $I = \{1,2,\ldots,C\}$, which may be a rectangular grid, an irregular tessellation, or operationally defined polygons used in mission planning. Probability maps over such segments are standard in search planning.

Hidden state (subject location). Let the (unknown) location of the subject at time t be a random variable $X_t \in I$. In the static subject case, $X_t = X_0$ for all t. In the moving subject case, X_t evolves according to a motion/transition model (Section 8).

3.2 Operational probabilities: POA (belief), POD (detectability), POS (tasking success)

1) POA / belief. At decision step t, the planning team maintains a belief distribution (probability map) $b_t(i)=P(X_t=i)$, with $\Sigma_i b_t(i)=1$.

2) POD. If sector i is searched using a specified resource/technique under stated conditions, the probability of detecting the subject conditional on the subject being in i is $p_i \in [0,1]$.

3) POS for a single tasking. If we task a search in sector i at time t, the probability of immediate success is $POS_t(i)=b_t(i)p_i$.

3.3 Actions (tasking decisions) and observations (found / not found)

At each decision step t, the search planner chooses an action $a_t \in I$, meaning "task a search in sector a_t." After executing the search, the team observes $O_t \in$ {Found, NotFound}. Because detection is imperfect, NotFound is not conclusive evidence of absence.

3.4 Observation model (imperfect detection made explicit)

3.4.1 Model contract (assumptions for the Bayes update)

To keep the belief update transparent and audit-friendly for SAR tasking, we adopt the following base observation model (extensions are noted in Appendix B8 and Section 13).

Sector-local information: a search tasking provides information only about the searched sector; other sectors change only through renormalization of POA.

Imperfect detection: if the subject is in the searched sector j, the search detects them with probability p(j) for that specific tasking (resource × time × conditions).

No false positives (base model): Found is not generated when the subject is not in the searched sector; ambiguous/partial observations are treated as an extension.

Operational note: many real searches can have cross-sector visibility or uncertain/partial clues. The base model is used here because it yields a simple, explainable Bayes update; richer observation models can be substituted without changing the overall POA/POD/POS workflow.

When a tasking provides information beyond the searched sector (e.g., line-of-sight from a ridgeline, clue drift, or sensor spillover), replace the sector-local likelihood with a non-local detection function $Z_j(i) = P(\text{Found} \mid X = i, a = j)$; the corresponding Bayesian update after Not-Found is given in **Proposition 3** (Appendix B8.1).

We adopt the standard imperfect-detection model:

• $P(O_t=\text{Found} \mid X_t=a_t, a_t)=p_{a_t}$.

• $P(O_t=\text{Found} \mid X_t \neq a_t, a_t)=0$.

Equivalently: $P(O_t=\text{NotFound} \mid X_t=a_t, a_t)=1-p_{a_t}$, and $P(O_t=\text{NotFound} \mid X_t \neq a_t, a_t)=1$.

Remark. When $p_{a_t}<1$, a negative result does not drive the posterior probability of that sector to zero; it reduces it—formalizing why re-search can be rational in low-detectability conditions.

3.5 Terrain and resource effects (where p_i comes from)

In practice, POD is not constant: it varies by terrain, weather, visibility, and resource type. We allow p_i to be sector-specific so terrain-aware and resource-aware detectability can be expressed directly and later modeled explicitly (Section 6).

3.6 Objective: what "optimal" means for SAR tasking

SAR planning goals are often described in two related ways:

1) Maximize probability of success within limited effort/time budget B.

2) Minimize expected time to detection E[T], where T is the first step at which Found occurs.

4. Bayesian Belief Updating and POS-Based Tasking

This section derives the Bayesian update after an unsuccessful search under imperfect detection and connects it to a transparent SAR tasking heuristic based on POS = POA × POD.

4.1 Bayes update after an unsuccessful search (Not-Found)

Notation for time indexing. Let $b_t(i)$ denote the planner's POA belief before tasking at decision step t, i.e., $b_t(i)=P(X_t=i \mid \text{history up to step } t)$. After executing the tasking at step t and observing the outcome, we write the posterior as $b_t^+(i)$. For static subjects ($X_{t+1}=X_t$), we carry $b_{t+1}(i)=b_t^+(i)$ into the next step.

Let the current belief be $b_t(i)=P(X_t=i)$. Suppose we search sector j (i.e., $a_t=j$) and observe Not-Found. Under the observation model: $P(\text{Not-Found} \mid X_t=j, a_t=j)=1-p_j$, and $P(\text{Not-Found} \mid X_t=i, a_t=j)=1$ for $i \neq j$.

Bayes' theorem gives the posterior after this observation:

$$b_t^+(i)=P(X_t=i \mid \text{Not-Found in } j)= P(\text{Not-Found} \mid X_t=i, a_t=j)\, b_t(i) \,/\, P(\text{Not-Found in } j).$$

The normalizing denominator is:

$$P(\text{Not-Found in } j)= (1-p_j)\, b_t(j) + \Sigma_{k \neq j}\, b_t(k) = 1 - p_j\, b_t(j).$$

Thus the posterior update is:

$b_t^+(j) = ((1-p_j)\, b_t(j)) / (1 - p_j\, b_t(j))$.

$b_t^+(i) = b_t(i) / (1 - p_j\, b_t(j))$ for $i \neq j$.

Static-subject carry-forward. When the subject is static, set $b_{t+1}(i) := b_t^+(i)$ and proceed to the next tasking decision. When the subject may move, Section 8 uses a predict–update cycle: propagate b_t^+ through a transition model to obtain the next prior, then apply the same Bayes update after each new search outcome.

4.1.1 Repeated unsuccessful searches of the same sector: closed form, clearance bounds, monotonicity, and sensitivity

Operational takeaway (re-search and clearance). In SAR tasking, repeated searches of the same sector are common—especially in low-detectability terrain. Under the imperfect-detection model, each additional Not-Found reduces (but does not eliminate) the sector's POA unless POD is perfect. What matters operationally is the rate of reduction: high-POD searches 'clear' a sector quickly, while low-POD searches only slowly reduce belief, making re-search and method substitution rational.

A planner-friendly way to see this is through odds. Let odds mean 'probability the subject is in the sector' divided by 'probability the subject is elsewhere.' Each Not-Found multiplies these odds by the miss probability $(1-p)$ for that tasking. This gives an intuitive clearance narrative suitable for briefings and re-tasking meetings; the exact closed form and clearance bounds are provided in Appendix D (Propositions 4–6).

Sensitivity to POD miscalibration. If POD estimates are off, the posterior can be too aggressive (overconfident clearance) or too conservative (unnecessary re-search). Sensitivity is often strongest early in the search and in low-POD terrain; hence POD calibration and documenting the basis of POD assumptions are operationally important. The derivative expression supporting this statement is given in Appendix D (Proposition 7; see the sensitivity derivation).

Planning note. If you need a concrete 'how many passes?' estimate for a clearance target, you can use the clearance-bound expression in Appendix D (Proposition 5) to translate a desired POA threshold into an approximate required number of re-searches at a given POD. This provides a principled way to justify continued searching (or a change of method) when terrain or conditions keep POD low.

Operational briefing phrasing. 'We searched this sector with POD p; a negative result reduces—but does not eliminate—the chance the subject is here. Because p is low/high, the reduction is modest/strong, so we should consider re-searching with a higher-POD resource or shifting effort according to updated POS.' This keeps the update interpretable while remaining faithful to Bayes' rule.

4.2 Operational interpretation: "clearance" depends on POD

Two consequences follow:

1) A negative result never clears an area completely unless POD is perfect. If $p_j<1$ and $b_t(j)>0$, then $b_{t+1}(j)>0$.

2) Higher POD produces stronger clearance for the same prior POA: as p_j increases, $b_{t+1}(j)$ decreases more sharply.

4.3 The POS tasking rule: maximize immediate probability of detection

The immediate probability of finding the subject by searching sector i at time t is $POS_t(i)=b_t(i)\, p_i$. Therefore, the simplest transparent tasking decision is: $a_t \in \arg\max_i (b_t(i)\, p_i)$.

4.4 Why the POS rule is a "myopic optimum" (and when it is enough)

Intuition on optimality. The POS-maximizing choice is exactly optimal in settings where (i) travel and switching costs are negligible or equal across sectors, (ii) each tasking consumes the same effort budget, and (iii) the only objective is immediate detection rather than information gathering for later. When these conditions do not hold (common in field SAR), POS remains a useful, auditable baseline, and the POMDP framing in Section 9 shows how to incorporate costs and lookahead without changing the belief-update logic.

The POS rule maximizes one-step success probability but does not explicitly plan multiple steps ahead (travel costs, future information gain, multi-asset scheduling). It remains valuable as an auditable, operationally interpretable baseline tightly coupled to Bayesian updates after unsuccessful searches.

4.4.1 A minimal optimality statement for POS

The POS tasking rule is often described as a myopic policy because it maximizes immediate probability of success rather than explicitly optimizing multi-step objectives. Under the base model, however, the rule is exactly optimal for the one-step objective of maximizing the probability of detection on the next tasking.

Proposition 1 (One-step optimality of POS). At decision step t with belief $b_t(i)=P(X_t=i)$ and sector-specific POD p_i for the chosen resource/technique, the action a_t that maximizes the probability of detecting the subject on the next tasking satisfies

$$a_t \in \arg\max_i b_t(i)\, p_i.$$

Equivalently, maximizing immediate detection probability is the same as maximizing $POS_t(i)=POA_t(i)\times POD(i)$. Detail proof is given in Appendix D(Proposition 1).

4.5 The complete SAR planning loop (task → update → re-task)

1) Start with a prior POA map $b_0(i)$.

2) At decision step t, compute $POS_t(i)=b_t(i)\, p_i$ for each sector i.

3) Task $a_t \in \arg\max_i POS_t(i)$.

4) If Found, terminate and report the location.

5) If NotFound, update to the posterior b_t^+ using Section 4.1 and renormalize.

6) Carry forward to the next step: for a static subject set $b_{t+1} := b_t^+$. For a moving subject, first predict the next prior using a motion model (Section 8), then continue.

4.5.1 Posterior after Found (Lemma 1)

Lemma 1. If a search of sector j returns Found, then $b^+(j)=1$ and $b^+(i)=0$ for all $i\neq j$.

Proof. Assuming the observation has nonzero model probability (e.g., $p_j>0$), Found can only be observed when X=j and the search is in j, hence the posterior mass collapses to j.

5. Algorithm and Implementation Details

This section presents an implementable SAR planning loop and explicitly corrects the flow logic: Found terminates and returns the location; Not-Found triggers a Bayesian update.

5.1 Overview: what the algorithm does in SAR terms

The algorithm maintains POA over sectors and uses POD to compute POS. After each tasking, it incorporates the outcome via Bayes' theorem and recomputes priorities for the next step.

5.2 Flowchart logic

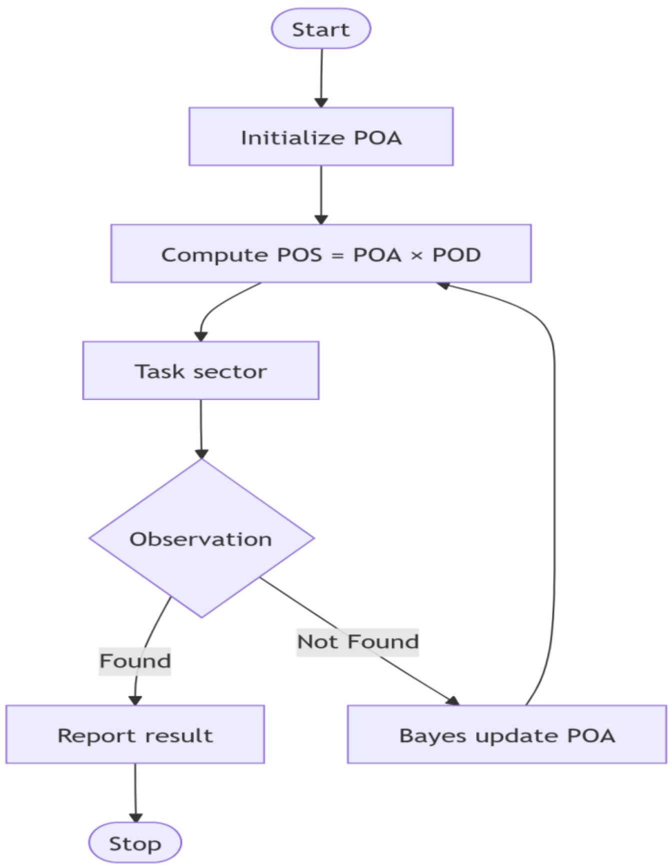


Figure 1. Bayesian SAR Tasking Loop.

5.3 Pseudocode (single-resource, sector-based tasking)

Inputs:

Sectors i = 1..C

Prior POA b0(i), sum_i b0(i) = 1

POD model p(i) in [0,1] for the chosen resource/technique

(Interpretation: p(i) is the Probability of Detection for that specific tasking—resource × time × conditions—not a permanent clearance value.) (1)

Termination condition: Found

Initialize:

b ← b0 // current prior POA map

Repeat for t = 0,1,2,... until Found:

1) For each sector i: POS(i) ← b(i) * p(i)

2) Choose sector j maximizing POS(i): j ← argmax_i POS(i) (apply operational tie-breakers if needed)

3) Execute search tasking in sector j and observe outcome O ∈ {Found, Not-Found}

4) If O == Found: return j // stop: subject located

5) If O == NotFound:

denom ← 1 - p(j)*b(j)

b_plus(j) ← (1 - p(j)) * b(j) / denom

For each i ≠ j: b_plus(i) ← b(i) / denom

b ← b_plus // carry posterior forward as next prior (static case)

continue

5.4 Implementation notes

Data structures: store POA vector b(i), POD layer(s) p(i) (possibly per resource), and sector metadata plus an evidence log.

Numerical stability and edge cases: if p(j)=0, Not-Found provides essentially no clearance; if p(j)=1, Not-Found clears the sector (b_{t+1}(j)=0). Clamp denominators against floating error.

Feasibility guard: Apply the Bayesian update only for observations with non-zero model probability. If 1 - p(j)b(j) = 0, then the event Not-Found in j has zero probability under the model; treat this as an assertion/logging condition and respecify p(j) (or review the observation) before updating.

Tie-breaking: document operational tiebreakers (travel time, safety/access, daylight, communications, etc.).

5.5 Multi-resource extension (practical SAR: POD depends on the asset)

Let resources r∈R and POD be p_r(i). Immediate POS becomes POS_t(i,r)=b_t(i)p_r(i). Choose (i*,r*) ∈ argmax_{i,r} POS_t(i,r). Bayesian updates use the POD of the resource that actually executed the search.

5.5.1 Sequential multi-asset searches of the same sector: effective POD
When multiple assets search the same sector, the Bayes update can be applied sequentially using the POD of each executed search. Under an independence assumption (conditional on the subject being present), this sequential updating is equivalent to a single update with an effective combined POD.

Proposition 2 (Effective POD for independent assets; sequential-update equivalence). Suppose sector j is searched by two assets (or two sequential taskings) with conditional detection probabilities p_1 and p_2, and suppose the detection events are independent given that the subject is in j. Then the conditional probability of failing to detect the subject after both searches is $(1-p_1)(1-p_2)$, so the effective combined POD is

$$p_{eff} = 1 - (1-p_1)(1-p_2).$$

Moreover, if both searches yield NotFound, applying the Bayes NotFound update sequentially (first with p_1, then with p_2) yields the same posterior as applying a single NotFound update with p_{eff}.

Detail proof of the above statement is provided in Appendix D(Proposition 2).

5.6 Logging and audit trail
Log per tasking: time/date, sector ID, resource/technique, assumed POD and basis, outcome Found/NotFound, and POA before/after update. This supports briefing, continuity, and after-action review.

5.7 Complexity
Per iteration: compute POS over all sectors O(C); update beliefs O(C); store belief map O(C).

6. Terrain-Aware POD (Probability of Detection) for SAR Tasking

A central message in operational search theory is that a negative search result is only as informative as the search's detectability: NotFound provides strong evidence only when POD is high. In practice POD varies with terrain, visibility, weather, access constraints, target type, and the search resource/technique. This section defines a terrain-aware POD layer that is interpretable, integrates with the Bayes update, and supports decisions about prioritization and resource selection.

6.1 Why terrain-aware POD matters
Terrain degrades detection through occlusion, line-of-sight limitations, and access constraints. Two sectors with similar POA can have different POS = POA×POD; a Not-Found in a low-POD sector should reduce POA far less than a Not-Found in a high-POD sector.

6.2 POD in SAR practice: from detection models to usable sector-level values
Operational search theory models POD via detection models tied to coverage, sweep width, and sensor performance. SAROPS uses sensor/platform functions (e.g., lateral range curves) and updates distributions after unsuccessful searches using Bayesian reasoning informed by detectability models. Land SAR field experiments also motivate condition-adjusted POD (5,6).

6.3 A simple, interpretable terrain-aware POD model (sector-level)
Let $p_r(i)$ be the POD for resource r in sector i. Model it as:

$$p_r(i) = \mathrm{clip}(p_{base\_r} \cdot \tau(i) \cdot \kappa(i) \cdot \omega(t), 0, 1).$$

Here p_{base_r} is a baseline POD for resource r; $\tau(i)$ is a terrain/cover factor; $\kappa(i)$ is an access/coverage factor; $\omega(t)$ is a time/condition factor (day/night, weather).

6.4 Terrain classes and example factors

Example terrain factors (illustrative only): open $\tau=1.0$; light forest $\tau=0.7$; dense forest $\tau=0.4$; steep/rocky $\tau=0.5$ with $\kappa<1$ due to reduced coverage; impassable/water $\kappa=0 \rightarrow p_r(i)=0$ for ground resources.

6.5 How terrain-aware POD changes decisions

Tasking uses POS ranking: choose $(i^*,r^*) \in \arg\max_{i,r} b_t(i)p_r(i)$. Belief updating uses the executed POD: high-POD searches produce stronger clearance, low-POD searches weaker clearance.

6.6 Resource-aware POD

Maintain multiple POD layers for different assets (ground, canine, UAV, helicopter). A dense sector can have high POA but low ground POD; assigning a different asset may increase POS and yield stronger evidence after Not-Found.

6.7 Practical estimation of terrain factors

Populate τ and κ from GIS land-cover, slope/access constraints, visibility/illumination, weather, and team expertise/debrief. Treat them as planning inputs and document their basis.

Section 10 and 11 give a Proof-of-concept and a worked example.

6.8 Limitations

Sector-level POD is simplified; operational deployment requires calibration and richer detection modeling (coverage/sweep width/sensor curves) (8,9).

7. Informed Priors (POA Initialization) from SAR Intelligence Inputs

Operational SAR planning rarely starts from a uniform prior. Planners synthesize investigative information into an initial probability map (prior POA/POC) guiding early tasking. SAR doctrine emphasizes vigorous investigation and probability maps; SAROPS formalizes priors via weighted scenarios. This section describes informed priors that are mathematically consistent, auditable, and compatible with Bayesian updating (1).

7.1 What is the "prior" in SAR terms?

The prior POA is $b_0(i)=P(X_0=i)$ with $\Sigma_i b_0(i)=1$.

7.2 Why informed priors matter

Early tasking is leverage under time pressure and spreading uncertainty. Bayesian posteriors depend on the prior; weak priors can misdirect early effort even with correct updates.

7.3 Sources of prior information

Datums (LKP/TLK), scenario-based priors (consistent stories), GIS/terrain features, evidence fusion (tracks/pings/sightings), and standardized methodology considerations.

7.4 Three practical ways to construct informed priors

Method A (weighted layers): $\tilde{b}_0(i)=\Sigma_k w_k L_k(i)$, then normalize $b_0(i)=\tilde{b}_0(i)/\Sigma_j \tilde{b}_0(j)$.

Method B (likelihood multiplication): $\tilde{b}_0(i)=b_{base}(i) \Pi_k P(E_k \mid X_0=i)$, then normalize.

Method C (scenario mixture): $b_0(i)=\Sigma_s \alpha_s b_0^{(s)}(i)$ with $\Sigma_s \alpha_s=1$.

7.5 Linking priors to tasking

First tasking uses $POS_0(i)=b_0(i)p(i)$. High POA but low POD may be de-prioritized or resourced differently; moderate POA but high POD may be prioritized.

7.6 A short hypothetical worked prior

With LKP proximity layer L1 and trail proximity layer L2, choose weights $w1>w2$: $\tilde{b}_0(i)=w1\,L1(i)+w2\,L2(i)$, then normalize.

7.7 Documentation and audit trail

Record which layers/scenarios were used, rationale for weights, and prior map version. This supports transparency and after-action review.

7.8 Limitations

Informed priors can encode bias; scenario mixtures and evidence tracking help. Bayes updates correct priors only if POD estimates are meaningful and outcomes are logged consistently.

8. Dynamic Subject Modelling (Moving Target / Drift) and Belief Propagation

Many SAR incidents involve moving subjects or drifting objects. Motion modeling is central to forming and updating probability maps; SAROPS treats object motion as a core component. We extend POA/POD to dynamic belief modeling: propagate POA forward using a motion model, then apply Bayes update after each search outcome (5,6).

8.1 Why dynamic modelling matters

Time affects both survivability and uncertainty; negative results age if the subject could move; planning should target where the subject is likely to be now.

8.2 Motion model as a transition matrix

Model movement with $T_{ij} = P(X_{t+1}=j \mid X_t=i)$, with $\Sigma_j T_{ij}=1$. Use identity for static, neighbor random walk for wandering, biased transitions for corridors/barriers, or drift-derived transitions for maritime contexts.

In WiSAR contexts, terrain-informed lost-person behavior models can help parameterize or validate these transition probabilities (e.g., terrain-feature-driven Bayesian/Markov models). (10)

8.3 Predict–update cycle

We use a predict–update cycle aligned with operational replanning epochs.

Prediction (propagate the posterior from the previous epoch): starting from $b_t^+(i)$, compute the next-epoch prior

$$\hat{b}_{t+1}(j) = \Sigma_i\, b_t^+(i)\, T_{ij}.$$

Tasking at epoch t+1: compute $POS_{t+1}(j)=\hat{b}_{t+1}(j)\, p(j)$ and choose the max-POS sector (or sector–resource pair).

Update after observing Not-Found in searched sector k at epoch t+1:

$$denom = 1 - p_k\, \hat{b}_{t+1}(k).$$

$$b_{t+1}^+(k) = ((1-p_k)\, \hat{b}_{t+1}(k)) / denom;$$

and for $j \neq k$, $b_{t+1}^+(j) = \hat{b}_{t+1}(j) / denom$.

Carry forward: set $b_{t+1} := b_{t+1}^+$ for the next prediction step.

8.4 Choosing the planning time step (Δt)

Choose Δt to match operational re-planning cadence (sorties, operational periods, major intel updates). Ensure transition probabilities reflect plausible movement within Δt.

8.5 How dynamic modelling changes decisions

Dynamic models support corridor/intercept searching, reinterpretation of “cleared” sectors as POA can return, and increased value of early high-POS tasking under spreading uncertainty.

8.6 Particles vs transitions

SAROPS uses particle simulation and Bayesian reweighting; the transition-matrix model is the sector-level analogue emphasizing transparency and lightweight computation (5,6).

8.7 Limitations and documentation

Transition models are uncertain and context-dependent; document assumptions, align time steps with operational cycles, and maintain auditability.

9. POMDP Framing for SAR Tasking

A Partially Observable Markov Decision Process (POMDP) provides a standard language for sequential decision-making under uncertainty and imperfect detection. This section anchors the POA/POD workflow in a POMDP and clarifies how richer planning objectives (travel cost, risk, multi-step lookahead, multi-asset assignment) can be incorporated without changing the core belief-update logic.

9.1 POMDP definition

A POMDP is specified by $\langle S, A, T, R, O, Z, \gamma \rangle$ with hidden states, actions, transitions, rewards/costs, observations, observation probabilities, and discount factor (13).

9.2 Instantiating the POMDP for sector-based SAR planning

State S: minimal state is subject sector $i \in \{1,\ldots,C\}$; optional extended state includes asset location for travel costs.

Actions A: search a sector j, or choose (j,r) for sector–resource assignments.

Transition T: identity for static; Markov transitions for moving subjects (Section 8).

Observations O: Found / Not-Found.

Observation model Z: $P(\text{Found} \mid s=i,a=j)=1[i=j]\, p_j$; $P(\text{NotFound} \mid s=i,a=j)=1-1[i=j]\, p_j$.

Reward/cost R: $+R_{find}$ on detection; otherwise negative search and travel costs.

9.3 Belief state b: POA as the sufficient statistic

Notation note: in this POMDP view, b_t is the belief (POA map) before action a_t, and b_t^+ is the updated belief after observing o_t; for static targets, $b_{t+1}=b_t^+$.

Belief update after action a and observation o: $b_{t+1}(i)=\eta\, Z(o \mid s=i,a)\, \Sigma_k T_{k i}\, b_t(k)$, with normalization η.

9.4 POS rule as a myopic policy

The POS tasking rule $a_t \in \operatorname{argmax}_j b_t(j)\, p_j$ is a myopic policy that maximizes immediate detection probability, while POMDP framing clarifies how to add lookahead and costs if needed.

9.5 Why not solve the full POMDP here?

Exact POMDP solutions are computationally intensive for large state/action spaces. In practice, SAR applications use approximations and transparent decision rules when interpretability and auditability are key.

9.6 Operational takeaway

POA maps are belief states; POD defines the observation model; Bayesian updating after Not-Found is the correct belief update; POS-based tasking is a transparent one-step policy; POMDP is the formal home for travel/risk/multi-step planning extensions.

Related applications. POMDP-style formulations similar to Section 9 have been used to support UAV search planning and victim-finding in disaster or humanitarian settings (14–16).

10. Simulation Study (Proof-of-Concept Demonstration)

This section provides a proof-of-concept simulation to illustrate core SAR planning behaviours: maintaining POA, selecting by POS, and updating POA after unsuccessful searches with a POD-dependent Bayes update. The simulation is illustrative, not operationally validated; limitations are stated and expanded in Section 13.

10.1 Purpose and scope

Purpose: demonstrate POS prioritization, POD-dependent clearance via Bayes updates, and terrain-aware POD effects. Not a deployment claim: operational performance requires calibrated POD models and workflow integration.

10.2 Environment and experimental setup

Use a discretized region of $C=M\times N$ sectors (e.g., 5×5 for visualization clarity). The framework is independent of grid size or geometry; operational sectors can be polygons.

10.2.1 Prior POA

Use either uniform prior $b_0(i)=1/C$ or an informed prior from Section 7.

10.2.2 Detection model (imperfect detection)

Each sector i has POD $p(i)\in[0,1]$. If the subject is in searched sector i, Found occurs with probability $p(i)$; otherwise Not-Found. If the subject is elsewhere, the outcome is Not-Found.

10.2.3 Terrain-aware POD scenarios

Compare uniform POD $p(i)=p_0$ with terrain-aware POD $p(i)=p_0\times\tau(i)\times\omega(t)$ as in Section 6.

10.3 Policies compared

Baseline clarification (imperfect detection). “Without replacement” refers only to the selection rule (do not choose the same sector twice) and should not be interpreted as a claim that a sector becomes fully cleared after one negative result when POD<1. This baseline is included only as a naive comparator to highlight the value of Bayesian belief updating under imperfect detection.

1) Random with replacement.

2) Random without replacement.

3) POA-only policy.

4) POS-myopic without posterior update (static plan).

5) Bayesian POS policy (this paper): choose argmax $b_t(i)p(i)$ and update via Bayes after Not-Found.

10.4 Evaluation metrics (SAR-relevant)
Report steps-to-detection T (mean/median/std), tail risk (90th/95th percentiles), probability of success within budget B: $P(T \le B)$, re-search frequency, and belief evolution visuals.

10.5 Experimental procedure (reproducible protocol)
Each trial: initialize b_0; sample X_0; repeat selection, simulate observation using imperfect detection with POD, update b via Bayes after Not-Found (if policy updates), stop on Found. Repeat for N trials and report metrics with uncertainty.

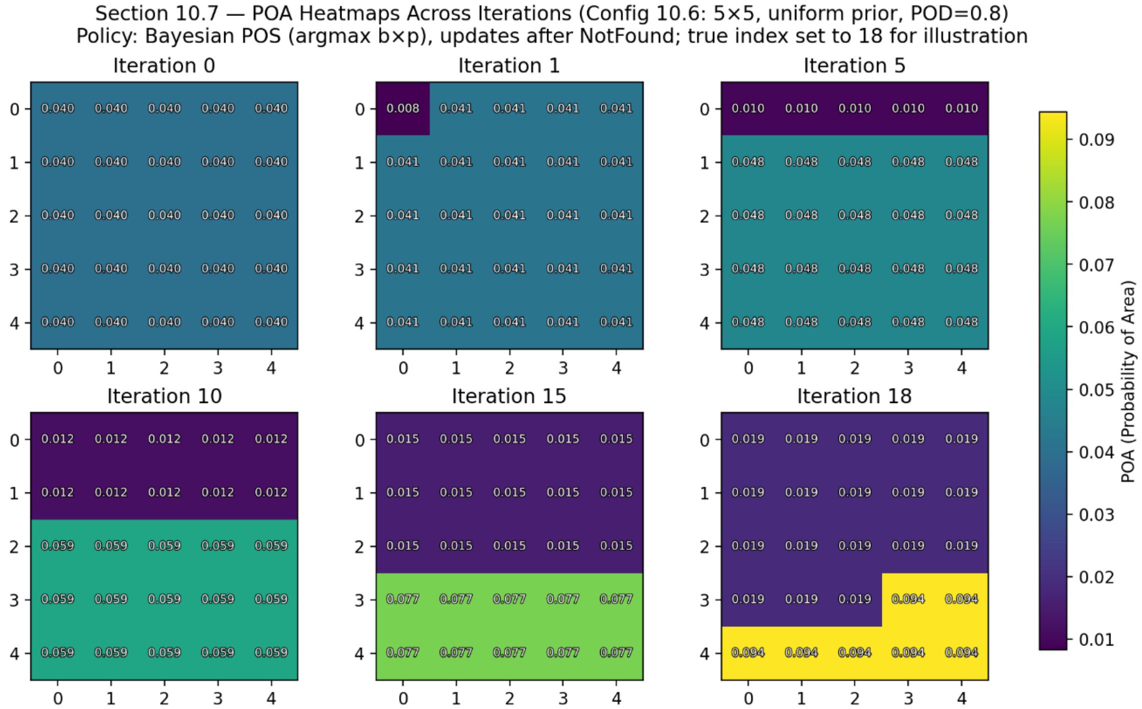
Implementation note: Found must be simulated as a Bernoulli event with probability $p(a_t)$ even when the searched sector equals the true sector; otherwise the simulation assumes perfect detection and contradicts the model.

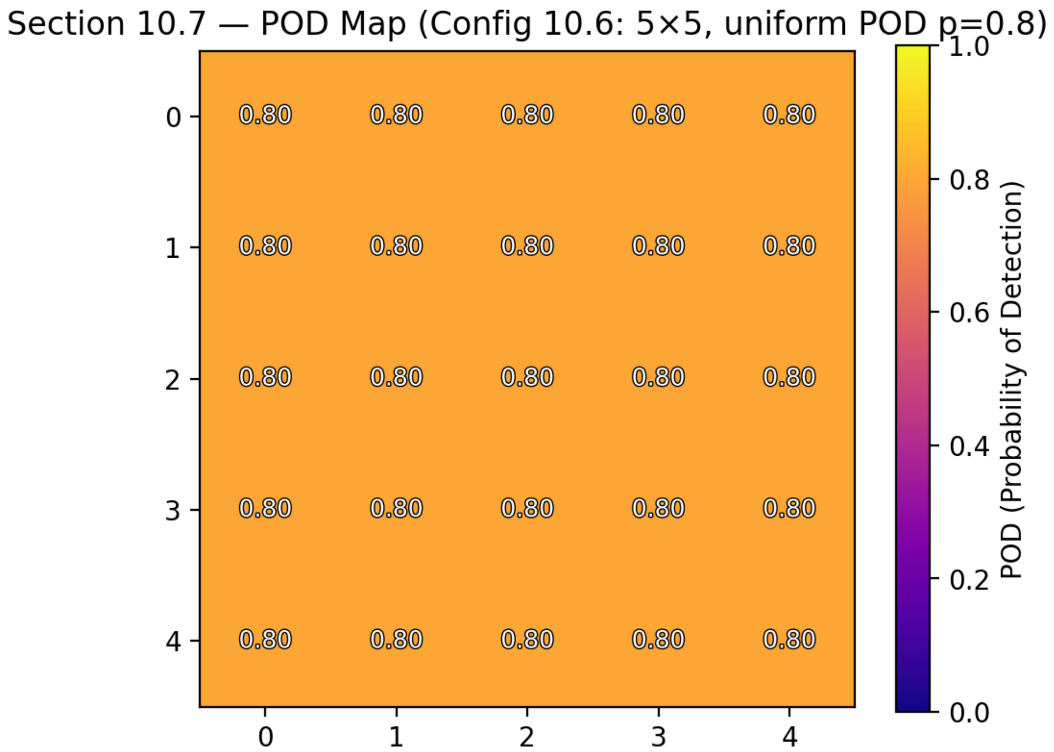
10.6 Results presentation
Configuration. 5×5 grid (C=25), **static** target, **uniform prior**, **uniform POD p=0.8**, **imperfect detection**, **N=5,000** trials per policy. Metrics: steps-to-detection mean/std, quantiles (T50, T90, T95), and budget success for . Policies: (P0) Random w/ replacement, (P1) Random w/o replacement, (P2) POA-only (with Bayes updates), (P3) POS-myopic static plan (no updates), (P4) Bayesian POS (with Bayes updates).

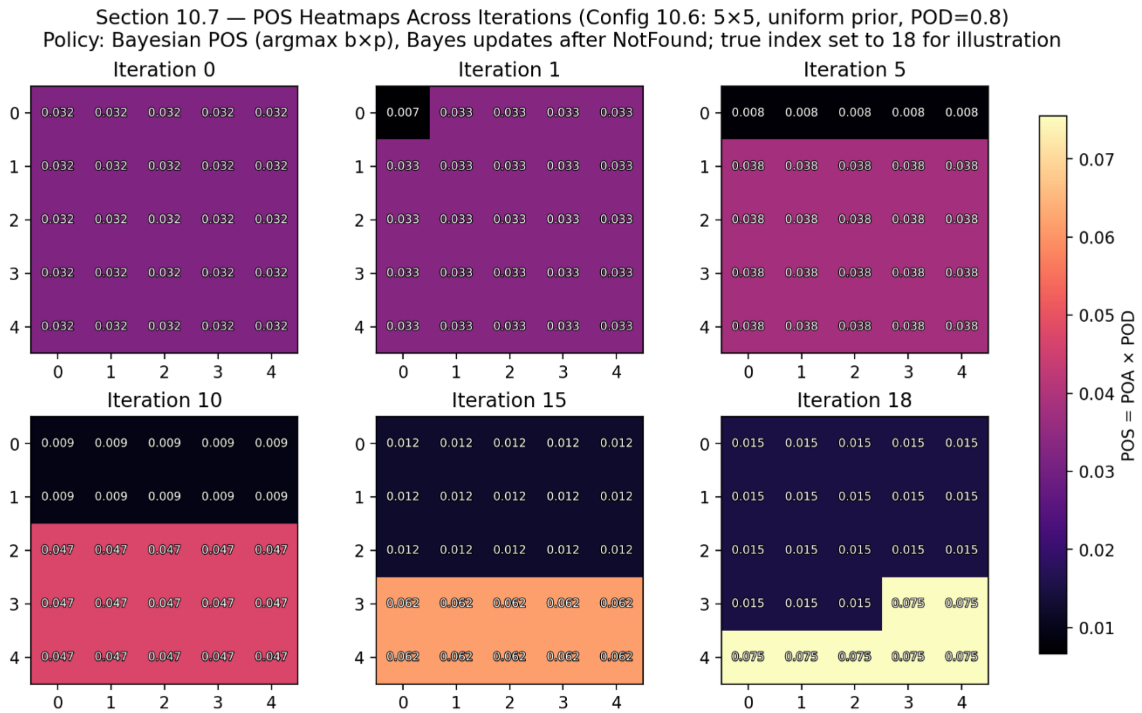
Policy	**Description**	**N**	**Mean_T**	**Std_T**	**T50**	**T90**	**T95**	**P(T≤5)**	**P(T≤10)**
P0	Random with replacement	5000	31.14	30.74	22.0	72.0	92.0	0.149	0.277
P1	Random without replacement	5000	19.39	15.36	16.0	41.0	49.0	0.147	0.308
P2	POA-only policy (with Bayes updates)	5000	19.53	16.14	16.0	42.0	50.0	0.158	0.316
P3	POS-myopic without posterior update (static plan)	5000	19.45	16.10	16.0	42.0	50.0	0.161	0.319
P4	Bayesian POS policy (with Bayes updates)	5000	19.17	15.63	16.0	41.0	49.0	0.171	0.321

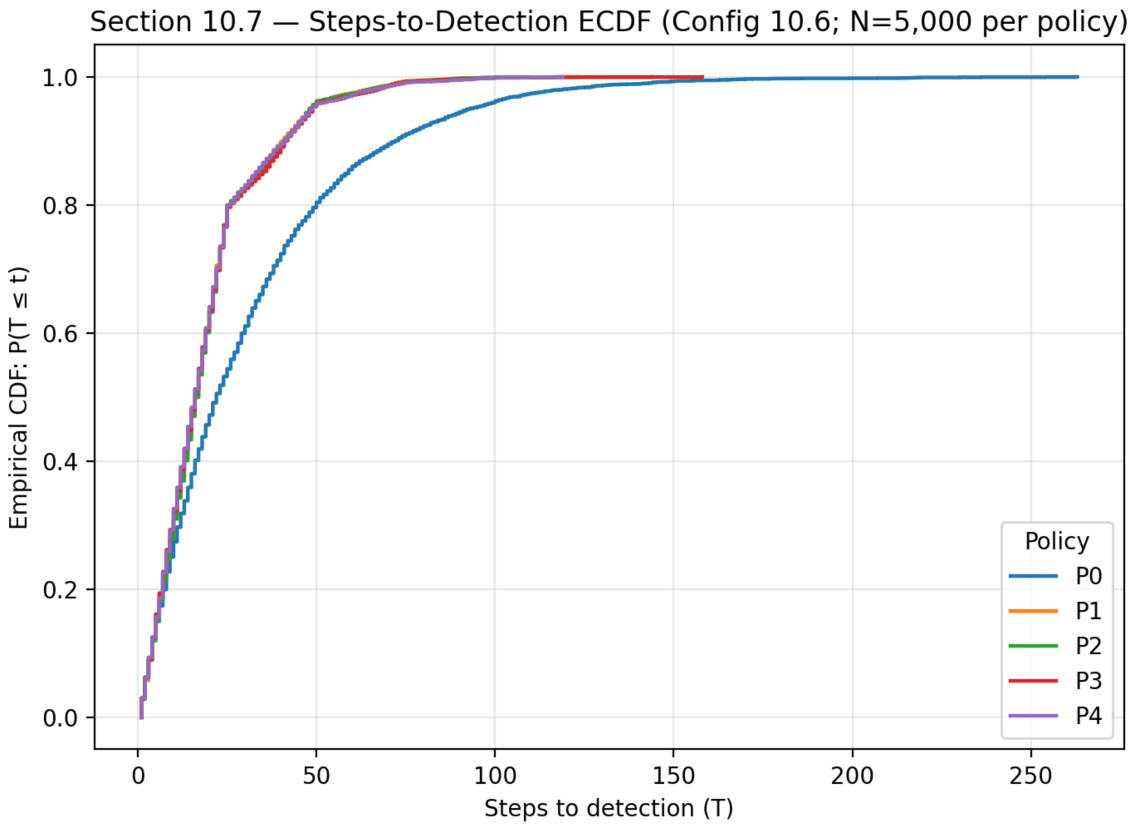
Qualitative belief concentration dynamics and quantitative metrics. [Numerical summary table after final aligned simulation runs, Python Pseudo code is given in Appendix C8]

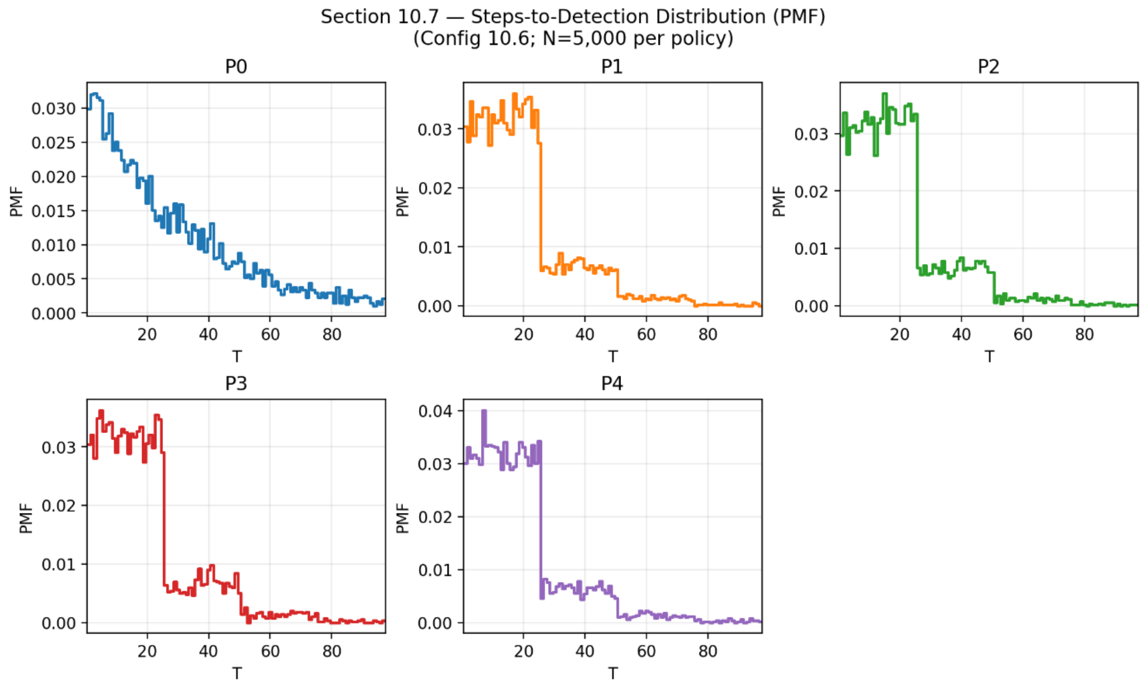
10.7 Figures: POA heatmaps across iterations, POD and POS maps, and steps-to-detection distribution plots.






Section 10.7 — POS Heatmaps Across Iterations (Config 10.6: 5×5, uniform prior, POD=0.8)
Policy: Bayesian POS (argmax b×p), Bayes updates after NotFound; true index set to 18 for illustration
Iteration 0
Iteration 1
Iteration 5
Iteration 10
Iteration 15
Iteration 18
POS = POA × POD


Section 10.7 — Steps-to-Detection ECDF (Config 10.6; N=5,000 per policy)
Empirical CDF: P(T ≤ t)
Steps to detection (T)
Policy
P0
P1
P2
P3
P4



10.8 Simulation limitations

Grid abstraction, simplified POD, single-asset simplification, and lack of mission-data validation limit operational inference; expanded in Section 13.

11. Worked Hypothetical SAR Example (How POA/POD/POS Drives Real Tasking)

This section provides a short worked example showing how POA/POD/POS tasking and Bayes updates would be used in practice. The scenario is hypothetical but mirrors SAR planning workflows.

11.1 Scenario description

A hiker is overdue in a mixed-terrain park. LKP is a trail junction. Terrain varies across sectors; resources include one ground team initially. Goal: prioritize the next two taskings and show how Not-Found updates POA and changes priorities.

11.2 Sectorization (simple 3×3 for clarity)

A	B	C
D	E	F
G	H	I

Sector E contains the LKP trail junction. A,B are more open; C,F,I are dense forest; G,H include steep or harder-access terrain.

11.3 Step 0: Build an informed prior POA

Illustrative prior: b_0(E)=0.20; b_0(B)=b_0(D)=b_0(F)=b_0(H)=0.10 each; corners A=0.05, C=0.02, G=0.02, I=0.01. Sum is 1.00.

11.4 Assign terrain-aware POD for a ground team

Assume daylight: open (A,B) p=0.70; mixed/trail (D,E,H) p=0.55; dense forest (C,F,I) p=0.30; steep/harder-access (G) p=0.35. Values are illustrative.

11.5 Compute POS and choose the first tasking

Compute POS_0(i)=b_0(i)p(i). Top candidates: E=0.110; B=0.070; D=0.055; H=0.055; A=0.035; F=0.030. Tasking 1: Search sector E.

11.6 Outcome: Not-Found in E → Bayesian update

With p_E=0.55 and b_0(E)=0.20, denom = 1 - 0.55×0.20 = 0.89. Updated b_1(E)=(0.45×0.20)/0.89 ≈ 0.1011. Other sectors scale by 1/0.89. POA in E does not go to zero because POD<1.

11.7 Recompute POS and choose the second tasking

Recompute POS_1(i)=b_1(i)p(i). Key values: B POS≈0.07865; E POS≈0.0556; D and H POS≈0.0618. Tasking 2: Search sector B (highest POS after update).

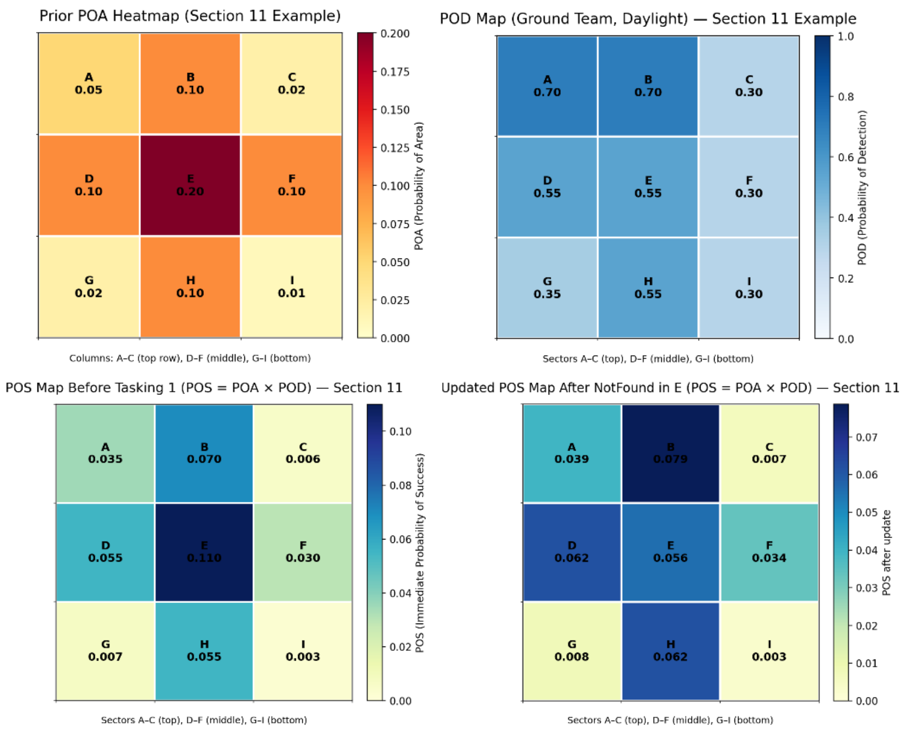
11.8 Why terrain-aware POD matters: a re-search insight

Dense forest sectors may retain nontrivial POA but have low ground POD, keeping POS modest and making negative results weak evidence. Operationally, consider resource substitution (canine/UAV/thermal) or planned re-search.

11.9 Briefing-friendly summary

Initial POA centered on LKP sector E; POD varies by terrain; task E first; NotFound partially clears E (POA decreases but does not vanish); recompute POS and task B next. Low-POD negatives should be interpreted cautiously.

11.10 Figure-ready artifacts : Prior POA heatmap, POD map, POS map before Tasking 1 and after Not-Found update.



12. Operational Implications for SAR Planning and Decision-Making

This section translates the framework into practitioner-facing implications: sector prioritization, allocation of effort, resource selection, re-search, evidence integration, and coordination.

12.1 Sector prioritization: turning POA and POD into an auditable tasking order

Rank sectors by POS = POA×POD and task the maximum-POS sector (or best sector–resource pair). This is explainable and aligns with operational search planning logic.

12.2 How belief updating changes the meaning of “Not Found”

Not-Found reduces POA in proportion to POD; sectors are not fully cleared unless POD=1. High-POD searches yield stronger clearance; low-POD searches yield weaker clearance and can justify re-search or resource substitution.

12.3 Allocation of effort across sectors

Use rolling-horizon planning: select a near-term set of high-POS sectors; update POA after each report; recompute priorities for responsive re-tasking.

12.4 Resource selection: matching assets to terrain to increase POD

Maintain $p_r(i)$ per resource. Assign assets to sectors where they maximize POS and produce meaningful evidence. This avoids false clearance under low POD and supports targeted deployment of specialized resources.

12.5 Re-search decisions

Re-search is rational when POA remains high after low-POD searches; less justified after repeated high-POD negatives. Prefer re-search with a better method that increases POD and information gain.

12.6 Integrating heterogeneous evidence into POA

Incorporate new clues mid-mission as POA updates (likelihood layers) or scenario reweighting; continue POS prioritization. This supports continuity and reduces planning churn.

12.7 Communication and coordination

POA/POD/POS provides a common language for briefings and handoffs, improving explainability, auditability, and after-action review.

12.8 Operational trade-offs made explicit

The framework makes explicit: thoroughness vs coverage, speed vs certainty, and access/safety constraints via POD layers and POS prioritization.

12.9 Deployment view

Success in deployment is a decision-support companion: updated POA map, explicit POD layer, and POS-ranked next-tasking list—integrated into planning cycles.

13. Limitations and Requirements for Operational Deployment

This paper’s simulations and examples are proof-of-concept demonstrations. Operational deployment requires data, calibration, modelling refinements, and validation. This section states limitations and outlines what is needed for operationalization.

13.1 Limitations of sector/grid abstractions

Grid cells are pedagogical; real sectors are irregular. Discretization and resolution trade-offs can hide corridors/barriers and within-sector heterogeneity.

13.2 Simplified detection models vs real POD estimation

POD is not a single constant; it depends on coverage, sweep width, visibility, target type, searcher skill, fatigue, navigation accuracy, and environment. The binary Found/Not-Found observation model assumes no false positives and sector-local information; richer observation models may be required (8,9).

13.3 Motion model limitations

Transition models are uncertain and profile/terrain/weather-dependent; time-step mismatch with operational cycles can distort propagation; scenario-based modelling is recommended.

13.4 Single-agent / simplified resource constraints

Real SAR is multi-resource and multi-agency; operationalization requires coordination, scheduling, and constraint handling (safety, communications, availability).

13.5 Limits of proof-of-concept simulations

Small synthetic simulations do not support direct operational conclusions; they illustrate dynamics. Operational claims require validated models and real data.

13.6 Requirements for operational deployment (data, modeling, validation)

13.6.1 Data requirements

• GIS-ready sectorization aligned to incident management.

• POD calibration sources (field experiments, sensor models, environmental inputs).

• Evidence inputs for informed priors (LKP/TLK uncertainty, tracks, pings, sightings, scenario weights).

• Motion model inputs (terrain-based behavior data or maritime drift models) if dynamic.

13.6.2 Modeling requirements

• Resource- and condition-specific POD models tied to coverage/sweep width or sensor curves (8,9).

• Richer observation models when ambiguous observations occur (false positives, partial clues).

• Time alignment with operational cycles and reporting cadence.

• Cost/constraint integration (travel, risk, safety) if needed, potentially via POMDP framing.

13.6.3 Validation requirements

• Retrospective case validation using archived missions and comparing prioritization.

• Prospective pilots in exercises with logging and evaluation.

• Sensitivity analyses for POD and prior uncertainty to build trust.

• Human factors/usability testing for planner adoption.

13.7 Summary

Provided here: a mathematically correct, operationally interpretable POA/POD/POS Bayesian workflow with terrain-aware POD, informed priors, and optional dynamics/POMDP framing. Needed for deployment: calibration, GIS integration, multi-asset coordination, and validation.

14. Conclusion

SAR search planning is fundamentally a problem of allocating limited effort under uncertainty: deciding where to task next, interpreting negative results under imperfect detection, and accounting for terrain and conditions that influence detectability and clearance. Operational search theory formalizes these ideas through POA/POC, POD, and POS (1).

This paper presented a SAR-native Bayesian decision-support workflow. We derived a Bayes update for unsuccessful searches under imperfect detection and a transparent POS-based prioritization rule (POA×POD) for tasking. The "vanishing posterior" insight—that POA does not collapse to zero when POD<1—provides a rigorous explanation for why re-search and resource substitution can be rational in low-detectability conditions.

We also outlined operationally relevant extensions: terrain- and resource-aware POD layers, informed priors from intelligence inputs and scenario thinking, motion models for moving subjects/drift, and a POMDP framing for longer-horizon planning with explicit costs and constraints. Proof-of-concept simulations and a worked example illustrate the update dynamics and planning interpretation. We explicitly acknowledged limitations and outlined requirements for operational deployment: calibrated POD models, GIS integration, scenario/evidence handling, multi-asset coordination, and validation against mission data or training exercises.

14.1 Future work (deployment-oriented)

Future work should prioritize: POD calibration for land SAR under varied conditions; GIS-integrated priors and scenario workflows; multi-asset coordination; and retrospective/prospective validation with human factors evaluation.

References

1. Frost JR. The Theory of Search: A Simplified Explanation. Soza & Company, Ltd. and Office of Search and Rescue, U.S. Coast Guard; October 1996. Available: https://navcen.uscg.gov/sites/default/files/pdf/Theory_of_Search.pdf

2. Koopman BO. Search and Screening (OEG Report No. 56). Operations Evaluation Group, Office of the Chief of Naval Operations, U.S. Navy Department; 1946. Available: https://www.informs.org/Explore/History-of-O.R.-Excellence/Documents/Bernard-O.-Koopman-Search-and-Screening-1946

3. Stone LD. Theory of Optimal Search. New York: Academic Press; 1975. (Mathematics in Science and Engineering, Vol. 118). ISBN: 978-0126724509.

4. Washburn AR. Search and Detection. Topics in Operations Research Series. 4th ed. Linthicum, MD: INFORMS; 2002. ISBN: 978-1877640179.

5. Kratzke TM, Stone LD, Frost JR. Search and Rescue Optimal Planning System (SAROPS). Metron, Inc. / U.S. Coast Guard Office of Search and Rescue; report date 2010-06-01 (DTIC ADA564779). Available: https://apps.dtic.mil/sti/citations/ADA564779 and https://www.metsci.com/wp-content/uploads/2019/08/Search-and-Rescue-Optimal-Planning-System.pdf

6. U.S. Coast Guard. Search and Rescue Optimal Planning System (SAROPS) Information Sheet. U.S. Coast Guard; PDF. Available: https://www.dco.uscg.mil/Portals/9/CG-5R/SARfactsInfo/SAROPSInforSheet.pdf

7. Cooper DC, Frost JR, Robe RQ. Compatibility of Land SAR Procedures with Search Theory. Prepared for U.S. Department of Homeland Security / U.S. Coast Guard Operations (G-OPR). Potomac Management Group, Inc.; 30 Dec 2003. Available: https://www.dco.uscg.mil/Portals/9/CG-5R/nsarc/LandSearchMethodsReview.pdf

8. Koester RJ. Land Search and Rescue Probability of Detection: New sweep widths values, correction factors, models, and detection model validation. Journal of Search and Rescue; 2020. Available: https://journalofsar.com/wp-content/uploads/2020/04/v4-7-Koester-POD-Syrotuck.pdf.pdf

9. Koester RJ, Chiacchia KB, Twardy CR, Cooper DC, Frost JR, Robe RQ. Use of the Visual Range of Detection to Estimate Effective Sweep Width for Land Search and Rescue Based on 10 Detection Experiments in North America. Wilderness & Environmental Medicine. 2014;25(2):132–142. doi:10.1016/j.wem.2013.09.016.

10. Lin L, Goodrich MA. A Bayesian approach to modeling lost person behaviors based on terrain features in Wilderness Search and Rescue. Computational and Mathematical Organization Theory. 2010;16:300–323. doi:10.1007/s10588-010-9066-2.

11. Rossmo DK, Velarde L, Mahood T. Optimizing Wilderness Search and Rescue: A Bayesian GIS Analysis. Journal of Search and Rescue; 2023. doi:10.61618/HWOQ8554. Available: https://journalofsar.com/optimizing-wilderness-search-and-rescue-a-bayesian-gis-analysis/

12. Nasar W, Da Silva Torres R, Gundersen OE, Karlsen AT. The Use of Decision Support in Search and Rescue: A Systematic Literature Review. ISPRS International Journal of Geo-Information. 2023;12(5):182. doi:10.3390/ijgi12050182.

13. Kaelbling LP, Littman ML, Cassandra AR. Planning and acting in partially observable stochastic domains. Artificial Intelligence. 1998;101(1–2):99–134. doi:10.1016/S0004-3702(98)00023-X. Available: https://people.csail.mit.edu/lpk/papers/aij98-pomdp.pdf

14. Bravo RZB, Leiras A, Cyrino Oliveira FL. The Use of UAVs in Humanitarian Relief: An Application of POMDP-Based Methodology for Finding Victims. Production and Operations Management. 2019;28(2):421–440. doi:10.1111/poms.12930. (Bibliographic record: https://econpapers.repec.org/RePEc:bla:popmgt:v:28:y:2019:i:2:p:421-440)

15. Villani PG, Cugnasca PS. A POMDP Approach to Map Victims in Disaster Scenarios. Logistics. 2024;8(4):113. doi:10.3390/logistics8040113. Available: https://www.mdpi.com/2305-6290/8/4/113

16. Zhang Y, Luo B, Mukhopadhyay A, et al. Shrinking POMCP: A Framework for Real-Time UAV Search and Rescue. arXiv:2411.12967 (cs.RO). 2024. doi:10.48550/arXiv.2411.12967. Available: https://arxiv.org/abs/2411.12967

17. International Maritime Organization (IMO). IAMSAR Manual (International Aeronautical and Maritime Search and Rescue Manual) overview page (joint IMO/ICAO publication). Available: https://www.imo.org/en/ourwork/safety/pages/iamsarmanual.aspx. See also: IAMSAR Manual Volume II (Mission Coordination), 2019 edition (ISBN 978-92-9258-806-9) sample

Appendix A. Practitioner Implementation Checklist (POA/POD/POS Bayesian Tasking)

This appendix is a field-usable checklist to implement the workflow in an operationally interpretable, auditable manner consistent with POA/POD/POS search planning logic.

A1. One-page concept map (what the tool does)

Inputs → Models → Outputs → Iteration

1) Inputs: sector map + initial POA + POD estimates + completed-search reports.

2) Models: POS ranking + Bayes update after Not-Found + optional motion propagation.

3) Outputs: updated POA map + POS-ranked tasking list + re-search/resource flags.

4) Iteration: re-run after each sortie/tasking debrief (plan → search → update → re-plan).

A2. Minimum viable implementation (MVI) checklist

A2.1 Sectorization

• Define taskable sectors (grid or polygons); assign IDs; record metadata (terrain class, access notes, hazards, size).

A2.2 Prior POA

• Choose $b_0(i)$ (sums to 1). If informed, document sources. If scenarios, record weights and rationale.

A2.3 POD per sector and resource

• For each resource r, assign $p_r(i)\in[0,1]$. Apply condition modifiers (day/night, weather). Label whether estimated/doctrine/calibrated.

A2.4 Tasking rule

• Compute $POS(i,r)=b(i)p_r(i)$. Task the (sector, resource) with highest POS (with tie-breakers).

A2.5 After-task update

• Record outcome Found/Not Found. Found → stop/report. Not Found → Bayes update using executed POD.

A3. Operational inputs (what planners typically already have)

Priors: LKP/TLK, tracks, pings, sightings, terrain corridors/barriers. Detectability: resource type/technique, day/night, visibility/weather, terrain/cover class.

A4. Outputs (what the tool should produce)

Core outputs: updated POA map, POD layer used, POS-ranked tasking list. Enhancements: re-search flags, evidence-strength note, probability-of-success within budget view.

A5. Audit trail (non-negotiable)

Log priors (layers/scenarios), each tasking's POD and outcome, POA snapshots before/after updates, and any mid-mission evidence injections and their effects.

A6. Quality control (QC) checklist

Check normalization $\Sigma_i\ b(i)=1$ after every update; bounds $b(i),p(i)\in[0,1]$; flow logic Found→stop, NotFound→update; clearance intuition (low POD → mild POA reduction).

A7. Optional modules

Dynamic subject module (belief propagation), scenario mixtures, and cost-aware planning via POMDP framing.

A8. Field card summary

Update POA with new intel → choose POD layer → compute POS → task highest POS → Found stop; NotFound update POA → repeat.

Appendix B. Notation and Assumptions (Compact Glossary + Model Contract)

This appendix collects notation and makes explicit the assumptions under which the Bayesian update and POS tasking are valid.

B1. Operational terminology mapping (SAR ↔ model)

Sector/segment: taskable unit. POA/POC: probability subject is in a segment. POD: detectability conditional on presence and search method/conditions. $POS=POA\times POD$ (1).

B2. Core symbols

C: number of sectors. $I=\{1,\ldots,C\}$. X_t: hidden location at step t. $b_t(i)=P(X_t=i)$. a_t: tasked sector. r: resource (optional). $p(i)$ or $p_r(i)$: POD. $O_t\in\{Found,NotFound\}$. $POS_t(i)=b_t(i)p(i)$. T: detection time.

B3. Observation model (imperfect detection contract)

If searching sector j and $X_t=j$, Found occurs with probability $p(j)$ and NotFound with $1-p(j)$. If $X_t\neq j$, Found has probability 0 and NotFound probability 1. (No false positives in base model.)

B4. Bayesian update after NotFound

$Denom=1-p(j)b_t(j)$. Update: $b_{t+1}(j)=((1-p(j))b_t(j))/denom$ and $b_{t+1}(i)=b_t(i)/denom$ for $i\neq j$.

B5. Policy definition

POS-based tasking: $a_t \in \mathrm{argmax}_i\, b_t(i)p(i)$. Resource-aware: $(i^*,r^*) \in \mathrm{argmax}_{\{i,r\}}\, b_t(i)p_r(i)$.

B6. Dynamic target notation

Transition matrix $T_{ij}=P(X_{t+1}=j \mid X_t=i)$. Prediction: $\hat{b}_{t+1}(j)=\Sigma_i\, b_t(i)T_{ij}$. Then apply Bayes update after Not-Found.

B7. Assumptions (model contract)

1) Single subject. 2) Subject in exactly one sector at a time. 3) Sectorization adequate for tasking/reporting. 4) Binary observations. 5) No false positives. 6) Known POD for that tasking. 7) Not-Found is sector-local information (others change by normalization). 8) Independent detection opportunities across time. 9) Task executed as assumed (POD reflects reality). 10) One sector search per update step (parallel searches require extension).

B8. Common extensions

B8.1 Non-local observations (cross-sector visibility / clue drift)

The base model assumes a search tasking provides information only about the searched sector (Section 3.4.1). In practice, some taskings provide partial information about neighboring sectors (e.g., line-of-sight from ridgelines, clue drift, or sensor spillover). This can be modeled by allowing the probability of observing Found to depend on the true sector i as well as the searched sector j.

Proposition 3 (Generalized Not-Found update for non-local observation models). Let $Z_j(i)$ denote the probability of observing Found when the true sector is i and the searched sector is j, i.e., $Z_j(i)=P(Found \mid X=i, a=j)$, with $0 \le Z_j(i) \le 1$. After searching sector j and observing NotFound, the posterior is

$b^+(i) = ((1 - Z_j(i))\, b(i)) / (\Sigma_k (1 - Z_j(k))\, b(k))$.

In the sector-local base model, $Z_j(i)=1[i=j]\, p_j$, and this reduces to the update in Section 4.1.

Proof is given in Appendix D3.

B9. Sanity checks

Normalization; clearance intuition; flow logic Found→stop, NotFound→update.

Appendix C. Reproducible Simulation Protocol (Baselines, Metrics, and Reporting Template)

This appendix specifies a reproducible simulation protocol ensuring consistency with imperfect detection, fair baselines, and SAR-relevant reporting.

C1. Goal and scope

Demonstrate POS-based tasking, Bayesian updating after Not-Found, and terrain-aware POD effects under controlled conditions. Not an operational calibration claim.

C2. Experiment design: factors, conditions, and seeds

Vary map size, prior type, POD type, and target dynamics. Record master seed and deterministic per-trial seeds. Recommend N≥1000 trials per condition for stable estimates; N=100 is acceptable for illustrative figures with clear labeling.

C3. Policies (baselines) — explicit definitions

P0: Random with replacement. P1: Random without replacement. P2: POA-only (with or without update, specified). P3: Static POS plan (no posterior update). P4: Bayesian POS (this paper) with Bayes updates.

C4. The imperfect detection simulation step (non-negotiable)

When searching the true sector, Found must occur with probability p(j) (Bernoulli). If the subject is elsewhere, the observation is Not-Found. Then apply Bayes update if the policy updates.

C5. Metrics (include academic and operationally interpretable ones)

Primary: steps to detection T. Tail: T50, T90, T95. Budget: P(T≤B). Diagnostics: re-search count, average POD used, optional POA entropy.

C6. Reporting standards

For each condition × policy: N, mean T, std, quantiles, P(T≤B) for two budgets, SE and optional 95% CI. State baseline fairness: same priors, POD maps, target sampling, and termination rule.

C7. Figure templates

Belief evolution heatmaps; POD layer maps; steps-to-detection distributions (CDF/boxplots). Captions must state relevance to sector prioritization, clearance strength, and tail risk.

C8. Configuration "contract" and Pseudocode

```
# Config
seed = 123456
rows, cols = 5, 5
C = rows * cols
N = 5000
b0 = [1/C] * C              # uniform prior (POA)
p  = [0.8] * C              # uniform POD
# Bayes update after NotFound in sector j
def bayes_update_notfound(b, j, p_j):
   denom = 1 - p_j * b[j]           # total prob of NotFound
   b_plus = b.copy()
   b_plus[j] = (1 - p_j) * b[j] / denom
   for i != j: b_plus[i] = b[i] / denom
   return b_plus
# Policy selectors
# P0: random w/ replacement              P1: random w/o replacement (per-cycle)
# P2: POA-only (argmax b), with updates     P3: POS static plan (no updates)
# P4: Bayesian POS (argmax b*p), with updates
def select(policy, b, p, state, rng):
```

```
    if policy == 'P0': return rng.randint(C), state
    if policy == 'P1':                    # walk a random permutation cyclically
        order, k = state or (rng.permutation(C), 0)
        j = order[k]; k = (k + 1) % C
        if k == 0: order = rng.permutation(C)
        return j, (order, k)
    if policy == 'P2': return argmax(b), state
    if policy == 'P3':
        order, k = state or (range(C), 0)     # fixed 0..C-1 cycling
        j = order[k]; k = (k + 1) % C
        return j, (order, k)
    if policy == 'P4': return argmax(b * p), state
# One trial
def run_trial(policy, rng):
    X = sample_state(b0, rng)  # true location
    b = b0.copy()
    state = None
    t = 0
    while True:
        j, state = select(policy, b, p, state, rng)
        t += 1
        if j == X and rng.bernoulli(p[j]) == 1:
            return t                    # Found
        else:
            if policy in ['P2', 'P4']:     # Bayesian update only for these
                b = bayes_update_notfound(b, j, p[j])
# Monte Carlo loop: run N trials for each policy, collect steps-to-detection T
# Then compute: mean/std, quantiles (T50/T90/T95), and budget success P(T≤5), P(T≤10).
``
```

C9. Reproducibility checklist

Confirm: imperfect detection is simulated correctly; baselines are explicit and fair; metrics include tail and budget success; code/config/seed are documented; figures are clean and captions SAR-relevant.

Appendix D. Detailed Proofs of Propositions 1–7

This appendix provides complete, step-by-step derivations for Propositions 1–7 stated in the manuscript. The proofs use only Bayes' theorem, algebra, and (where convenient) odds transformations; they are included for completeness and to support auditability.

D.1 Proof of Proposition 1 (One-step optimality of POS)

Proof. If sector i is tasked next, then under the observation model the probability of Found on that tasking equals $P(X_t=i)\cdot P(\text{Found} \mid X_t=i,a_t=i)=b_t(i)\, p_i$. Therefore, choosing i that maximizes $b_t(i)\, p_i$ maximizes the one-step detection probability. This one-step optimality does not imply global optimality for multi-step objectives when travel costs, risk, time-varying POD, or information-gathering value are significant; the POMDP framing in Section 9 provides the formal setting for such longer-horizon trade-offs.

D.2 Proof of Proposition 2 (Effective POD and equivalence)

Proposition 2 restated. Two independent searches of the same sector j with PODs p_1 and p_2 have effective POD $p_{eff} = 1 - (1-p_1)(1-p_2)$, and two sequential NotFound Bayes updates are equivalent to one NotFound update with p_{eff}.

Proof. (a) Effective POD: conditional on the subject being in j, each search fails with probabilities $(1-p_1)$ and $(1-p_2)$. Independence implies joint failure probability $(1-p_1)(1-p_2)$. Thus detection probability over the two searches is $p_{eff} = 1 - (1-p_1)(1-p_2)$.

(b) Equivalence: let $r=b/(1-b)$ be odds for sector j. A NotFound update with POD p multiplies odds by $(1-p)$ (see Proposition 4 below). Two NotFound updates multiply by $(1-p_1)(1-p_2)$. A single NotFound update with p_{eff} multiplies odds by $1-p_{eff}=(1-p_1)(1-p_2)$. Hence posteriors coincide.

D.3 Proof of Proposition 3 (Generalized Not-Found update for non-local observation models)

Proof. Fix a searched sector j and let $Z_j(i)=P(\text{Found} \mid X=i, a=j)$ denote the probability of observing Found when the true sector is i and the action searches j. Under this observation model, the likelihood of NotFound is

$P(\text{NotFound} \mid X=i, a=j) = 1 - Z_j(i)$.

Let $b(i)=P(X=i)$ be the prior belief before searching j. By Bayes' theorem, after observing NotFound we have

$b^+(i)=P(X=i \mid \text{NotFound}, a=j) = [P(\text{NotFound} \mid X=i, a=j) * b(i)] / [\sum_k P(\text{NotFound} \mid X=k, a=j) * b(k)]$.

Substituting $P(\text{NotFound} \mid X=i, a=j)=1-Z_j(i)$ gives

$b^+(i) = [(1-Z_j(i)) * b(i)] / [\sum_k (1-Z_j(k)) * b(k)]$,

which is the claimed update. In the sector-local base model, $Z_j(i)=1[i=j]\, p_j$, so the expression reduces to the update derived in Section 4.1.

D.4 Proof of Proposition 4 (Repeated NotFound; closed form)

Proposition 4. Fix a sector j with constant tasking-specific POD $p_j = p \in [0,1]$. Suppose sector j is searched k times and the outcome is NotFound each time. Then: (a) $b_k(j) = b_0(j)\, (1-p)^k / ((1-b_0(j)) + b_0(j)\, (1-p)^k)$; (b) for $i \neq j$, $b_k(i) = b_0(i) / ((1-b_0(j)) + b_0(j)\, (1-p)^k)$.

Proof. Let $b(j)$ denote the POA of sector j before a search of j, and let $q = 1-p$. Under the base observation model, when j is searched and NotFound is observed, Bayes' theorem gives the one-step update (Section 4.1):

$b^+(j) = (q\, b(j)) / (1 - p\, b(j)) = (q\, b(j)) / ((1-b(j)) + q\, b(j))$.

Define the odds of sector j as $r = b(j) / (1 - b(j))$. We compute how r transforms under the update:

$r^+ = b^+(j) / (1 - b^+(j))$.

Substitute $b^+(j) = (q\, b)/((1-b) + q\, b)$:

$1 - b^+(j) = 1 - (q\, b)/((1-b) + q\, b) = ((1-b) + q\, b - q\, b)/((1-b) + q\, b) = (1-b)/((1-b) + q\, b)$.

Therefore

$r^+ = [(q\, b)/((1-b) + q\, b)] / [(1-b)/((1-b) + q\, b)] = q\, b/(1-b) = q\, r$.

So each NotFound update in sector j multiplies the odds by q. After k consecutive NotFound outcomes,

$r_k = q^k\, r_0 = (1-p)^k \cdot [b_0(j)/(1-b_0(j))]$.

Convert odds back to probability using $b_k(j) = r_k/(1+r_k)$:

$b_k(j) = [q^k b_0(j)/(1-b_0(j))] / [1 + q^k b_0(j)/(1-b_0(j))]$

$= [b_0(j) q^k] / [(1-b_0(j)) + b_0(j) q^k]$

$= b_0(j) (1-p)^k / ((1-b_0(j)) + b_0(j) (1-p)^k)$, proving (a).

For $i \neq j$, the base model implies P(NotFound | X=i, a=j)=1, so Bayes' rule gives $b^+(i)=b(i)/\text{denom}$ where denom = $1 - p\, b(j)$. Under repeated NotFound searches of j with constant p, the same normalization factor after k steps is $((1-b_0(j)) + b_0(j) (1-p)^k)$. Because all $i \neq j$ scale by the same factor and the distribution renormalizes,

$b_k(i) = b_0(i) / ((1-b_0(j)) + b_0(j) (1-p)^k)$, proving (b).

D.5 Proof of Proposition 5 (Clearance bound)

Proposition 5. Assume $0 < p < 1$ and $0 < \varepsilon < b_0(j)$. Let $b_k(j)$ be as in Proposition 4. Then the smallest integer k such that $b_k(j) \leq \varepsilon$ satisfies $k \geq \log(\varepsilon (1-b_0(j)) / (b_0(j) (1-\varepsilon))) / \log(1-p)$.

Proof. Start from the closed form in Proposition 4 with $q=1-p$:

$b_k(j) = b_0(j) q^k / ((1-b_0(j)) + b_0(j) q^k)$.

Solve $b_k(j) \leq \varepsilon$. Multiply both sides by the positive denominator:

$b_0(j) q^k \leq \varepsilon[(1-b_0(j)) + b_0(j) q^k] = \varepsilon(1-b_0(j)) + \varepsilon\, b_0(j) q^k$.

Bring q^k terms to the left and factor:

$b_0(j)(1-\varepsilon) q^k \leq \varepsilon(1-b_0(j))$.

Divide by the positive quantity $b_0(j)(1-\varepsilon)$:

$q^k \leq \varepsilon(1-b_0(j)) / (b_0(j)(1-\varepsilon))$.

Since $0<q<1$, $\log(q)=\log(1-p)<0$. Taking logs and dividing by $\log(q)$ reverses the inequality direction, yielding the stated lower bound on k.

D.6 Proof of Proposition 6 (Monotonicity)

Proposition 6. Fix $b_0(j) \in (0,1)$. For any $k \geq 1$: (i) $b_k(j)$ is strictly decreasing in p on (0,1); (ii) for fixed $p \in (0,1)$, $b_k(j)$ is strictly decreasing in k; (iii) if $p<1$ then $b_k(j)>0$ for finite k, while if $p=1$ then $b_1(j)=0$.

Proof. Let $q=1-p$ and write $b_k(j)=f(q^k)$ with $f(x)=b_0(j)x/((1-b_0(j))+b_0(j)x)$. Differentiate:

$f'(x)= b_0(j)(1-b_0(j)) / ((1-b_0(j)) + b_0(j)x)^2 > 0$ for $x>0$, so f is strictly increasing.

For fixed $k\geq1$, q^k strictly decreases in p on (0,1), hence $b_k(j)=f(q^k)$ strictly decreases in p. For fixed $p\in(0,1)$, $q\in(0,1)$ so q^k strictly decreases in k, hence $b_k(j)$ strictly decreases in k.

If $p<1$ then $q>0$ so $q^k>0$ for finite k and thus $b_k(j)>0$. If $p=1$ then $q=0$ so for $k\geq1$, $b_k(j)=f(0)=0$; in particular $b_1(j)=0$.

D.7 Proof of Proposition 7 (Sensitivity to POD miscalibration)

Proposition 7. Under Proposition 4 with $0<p<1$,

$\partial b_k(j)/\partial p = - k\cdot b_0(j)\cdot(1-b_0(j))\cdot(1-p)^{k-1} / (((1-b_0(j)) + b_0(j) (1-p)^k)^2) < 0$,

and therefore

$|\partial b_k(j)/\partial p| = k\cdot b_0(j)\cdot(1-b_0(j))\cdot(1-p)^{k-1} / (((1-b_0(j)) + b_0(j) (1-p)^k)^2)$.

Proof. Let $b0=b_0(j)$, $q=1-p$, and $D(q)=(1-b0)+b0\, q^k$. Then $b_k(j)=b0\, q^k / D(q)$. Differentiate with respect to q:

$d/dq\, [b0\, q^k / D] = (b0\, k\, q^{k-1}\cdot D - b0\, q^k\cdot D')/D^2$.

Compute $D'(q)=b0\, k\, q^{k-1}$. Substitute and simplify:

$d\, b_k/dq = b0\, k\, q^{k-1}(D - b0\, q^k)/D^2 = b0\, k\, q^{k-1}(1-b0)/D^2$.

Since $q=1-p$, $dq/dp=-1$, so $\partial b_k/\partial p = -(d\, b_k/dq)$. Taking absolute values and substituting $q=1-p$ yields the stated expression.