Coordinating UGV Fleets under Epistemic Uncertainty

A Dynamic-Epistemic-Logic Formulation of the Planning Problem, with a Scaling Benchmark

Haniel Ulises

Epistemic Robotics

August 12, 2026

Abstract. Fleets of unmanned ground vehicles that sense locally and communicate imperfectly face a coordination problem that is not captured by classical planning over world states: what has to be achieved is a configuration of *knowledge* across the fleet, and the actions that achieve it (sensing, driving into radio range, relaying a report) are valuable only for their epistemic effect. This report formulates that problem in Dynamic Epistemic Logic (DEL), where an epistemic state is a Kripke model and an action is an event model applied by product update, and it states the resulting planning problem: from a multi-pointed initial model, find a policy whose branches are the designated events of sensing actions and whose leaves all satisfy a modal goal formula. Six coordination scenarios are described conceptually and encoded as four EPDDL domains: facility-wide broadcast, range-limited radio, second-order goals, deployment geometry, addressed radio with selective-disclosure goals, and a heterogeneous fleet whose capabilities interlock. A benchmark of 204 instances, scaling the fleet size, the facility size, the amount of initial uncertainty, and the number of items to localise, is grounded with the plank toolkit and solved with the aletheia planner; 196 are solved under a 60 s budget. The measurements separate two costs that are easily conflated: grounding scales with fleet size times facility size, while search is dominated by the number of worlds the fleet initially considers possible, three orders of magnitude across the grid, against one for fleet size. Restricted communication costs a further factor of three and accounts for every unsolved instance. The suite also isolates a reproducible correctness fault in the planner, on goals conjoining knowledge requirements over several atoms.

Contents

1 Introduction

A team of unmanned ground vehicles (UGVs) inspecting a facility rarely fails because its members cannot compute a route. It fails because one vehicle knows something the others do not, and no one has established who knows what. A vehicle that has scanned a bay and found the target holds information that is useless until it is transmitted; a vehicle that transmits out of radio range has changed nothing; a vehicle that must act only once it is sure a teammate has received a report needs to reason about the teammate's knowledge, not just about the world.

Classical planning has no vocabulary for this. Its state is an assignment to propositions, and its goal is a condition on that assignment. It can express *the target is in bay 4*; it cannot express *every vehicle knows which bay the target is in*, still less R_1 *knows that* R_2 *knows*. Contingent and conformant planning add uncertainty but keep the perspective single-agent: they model what *the planner* does not know, not what the individual vehicles know about each other.

Dynamic Epistemic Logic (DEL) supplies exactly the missing vocabulary. A state of affairs is a Kripke model whose worlds are the situations the agents consider possible and whose accessibility relations encode, per agent, which situations that agent cannot tell apart. An action is a second Kripke-style structure, an event model, whose events carry preconditions and postconditions and whose relations encode who observes what. Executing an action is the product update of the two structures. Because observation is part of the action's definition, the same physical act can be public, private, or unobserved depending only on the observability relation attached to it, and that distinction is precisely the one that governs UGV coordination.

This report has three purposes, and deliberately stops short of a fourth. It (i) states the UGV coordination problem as a DEL planning problem; (ii) describes, conceptually, four scenario families in which the epistemic structure of the problem changes qualitatively; and (iii) reports a scaling benchmark that measures where a current DEL planner stops being practical on them. It does *not* address execution, ROS 2 integration, controller synthesis, or perception; those concerns are orthogonal to the planning problem treated here and are the subject of a companion document.

2 Background: the planning problem

2.1 Epistemic language

Fix a finite set $\mathcal{A}$ of agents (the vehicles) and a finite set $\mathcal{P}$ of atomic propositions. The language $\mathcal{L}$ is

$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid K_i\varphi \mid C_G\varphi, \qquad p \in \mathcal{P},\ i \in \mathcal{A},\ G \subseteq \mathcal{A}.$$

$K_i\varphi$ reads *agent* i *knows* φ and $C_G\varphi$ φ *is common knowledge in* G. Two derived operators carry most of the weight in what follows:

$$Kw_i\varphi \ \equiv\ K_i\varphi \vee K_i\neg\varphi, \qquad\qquad \widehat{Kw_i}\varphi \ \equiv\ \neg Kw_i\varphi.$$

$Kw_i\varphi$, i *knows whether* φ, is the natural form of an information-gathering goal: the fleet is not asked to make the target be in a particular bay, only to find out which bay it is in. Its dual states ignorance, and is how an initial state says that nobody knows yet.

2.2 Epistemic states and actions

Definition 1 (Epistemic state)**.** An epistemic state is a multi-pointed Kripke model $\mathcal{M} = (W, \{R_i\}_{i\in\mathcal{A}}, V, W_d)$ where W is a finite set of worlds, $R_i \subseteq W \times W$ is agent i's indistinguishability relation, $V : W \to 2^{\mathcal{P}}$ is a valuation, and $\emptyset \neq W_d \subseteq W$ is the set of *designated* worlds, those that are candidates for the actual one. Knowledge is modelled in S5: each R_i is an equivalence relation.

The size of W_d is the modeller's statement about what is objectively open. If the true bay of the target is asserted in the initial state, $|W_d| = 1$ and the planner may exploit knowledge the vehicles do not have; leaving it open gives $|W_d| > 1$ and forces a plan that works for every candidate. This distinction is not cosmetic, Section 5 keeps every benchmark instance in the open setting for exactly this reason.

Definition 2 (Event model)**.** An action is an event model $\mathcal{E} = (E, \{Q_i\}_{i\in\mathcal{A}}, pre, post, E_d)$: E is a finite set of events, $Q_i \subseteq E \times E$ says which events agent i cannot distinguish, $pre : E \to \mathcal{L}$ gives each event an epistemic precondition, *post* assigns literal postconditions, and $E_d \subseteq E$ are the designated events, the outcomes that may actually occur.

Definition 3 (Product update)**.** $\mathcal{M} \otimes \mathcal{E}$ has worlds $\{(w, e) : \mathcal{M}, w \models pre(e)\}$, relation $(w, e)\ R_i'$ (v, f) iff $w\ R_i\ v$ and $e\ Q_i\ f$, valuation revised by $post(e)$, and designated worlds $\{(w, e) : w \in W_d,\ e \in E_d\}$.

The observability relation Q_i is the whole modelling story. An event model whose events are all Q_i-distinguishable for every i is a public action; if some agent relates every event to a distinguished *null* event, that agent is *oblivious*, unaware not only of the outcome but of the occurrence. Obliviousness is the feature that makes short-range radio interesting: a vehicle out of range does not know that a report was sent, so it cannot even infer that its teammates are now better informed.

2.3 The planning task

Definition 4 (Epistemic planning task)**.** A task is $\Pi = (\mathcal{M}_0, \mathcal{A}, \gamma)$ with $\mathcal{M}_0$ an initial epistemic state, $\mathcal{A}$ a finite set of ground actions (event models), and $\gamma \in \mathcal{L}$ a goal formula. An action a is applicable in $\mathcal{M}$ when every designated world of $\mathcal{M}$ satisfies *pre* of some designated event of a.

A sensing action has more than one designated event, one per possible reading, and the executing agent learns which one occurred. A solution is therefore not a sequence but a *policy*, naturally represented as an AND/OR tree: OR-nodes choose an action, AND-nodes branch over the designated events of that action, and every leaf must satisfy γ. When no action in a plan has more than one designated event, the tree degenerates to a sequence and the task is a sequential epistemic planning problem.

Two properties of this problem shape everything downstream. First, plan existence is undecidable in general: unrestricted event models with epistemic preconditions can encode arbitrary computation, and decidability is recovered only under restrictions such as bounding modal depth or forbidding postconditions. Second, even in the decidable fragments the state representation grows multiplicatively, $|W|$ after k product updates is bounded by $|W_0| \cdot \prod_{j\le k} |E_j|$, so the practical limit is set by how quickly the model grows along the search frontier, not by the length of the plan.

2.4 From specification to search: the toolchain

The scenarios below are written in EPDDL, a PDDL-style specification language for DEL tasks in which one declares events with epistemic preconditions, reusable *action types* (public/private/semi-private ontic, sensing, and announcement templates), and per-agent observability conditions that may themselves be formulas. The plank toolkit parses, type-checks, and grounds a specification into an explicit JSON task; the aletheia planner reads that task and searches. Figure 1 shows the pipeline used throughout.

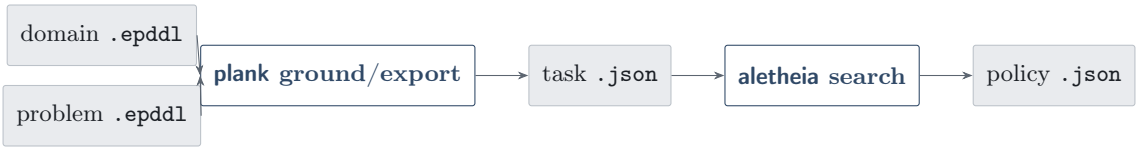

Figure 1: Specification-to-policy pipeline. Grounding is where the fleet size and corridor length turn into ground actions; search is where the epistemic model grows.

3 From robot software to epistemic models

The formalism of Section 2 talks about worlds and accessibility relations; a UGV stack talks about occupancy grids, pose estimates, sensor footprints and link budgets. This section states the correspondence, because the epistemic planning problem is only well posed once it is fixed.

3.1 Metric map to symbolic state

The navigation layer maintains a metric occupancy grid: a discretisation of the facility into cells, each free, occupied, or unknown. Planning over that grid directly is hopeless for an epistemic planner, a 20×6 grid already gives 2^{120} valuations before any modality is added. The standard remedy is the standard one in task-and-motion architectures: quotient the free space into a small set of *regions* whose adjacency graph is what the task layer reasons about. Here the regions are bays, obtained from the shelving structure, and the adjacency relation $\texttt{(adjacent b}_i\ \texttt{b}_j\texttt{)}$ is a static fact of the EPDDL problem. Figure 2 shows the quotient.

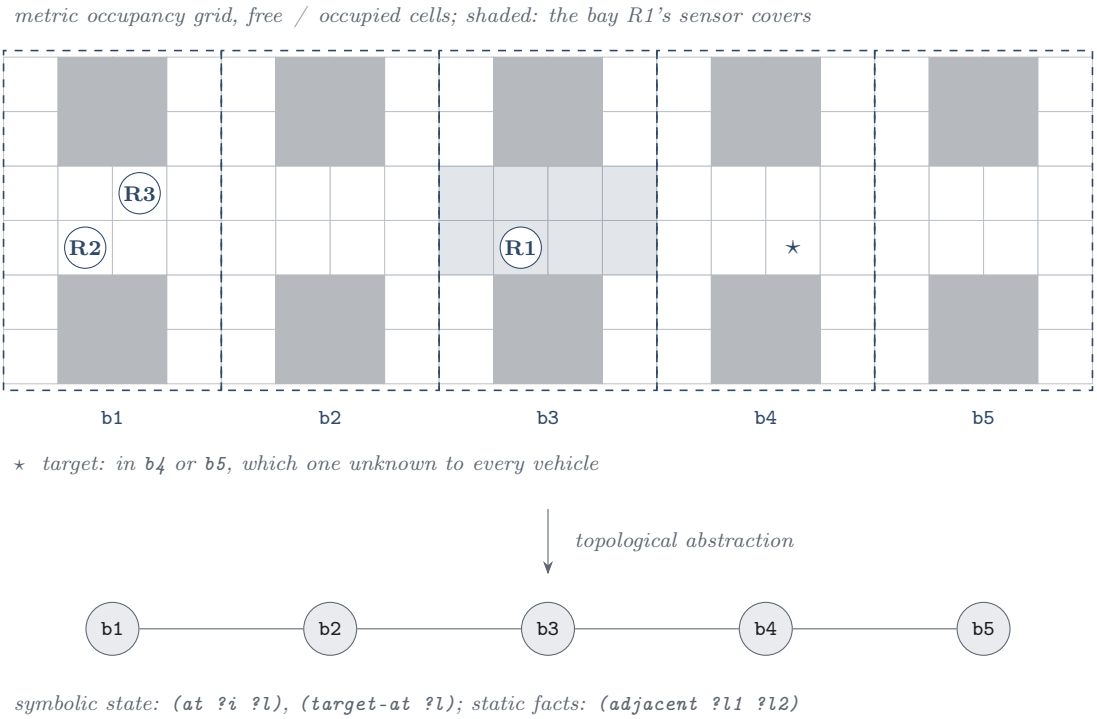

Figure 2: The abstraction the task layer plans over. Cells are quotiented into bays; the sensor footprint is assumed to cover exactly the occupied bay, which is what licenses `scan` being a bay-local sensing action; the adjacency graph of bays becomes static facts in the EPDDL problem.

Three modelling commitments are hidden in that picture, and each is an assumption the robotics layer must actually discharge.

1. **Navigation is assumed to succeed.** `move` is a deterministic ontic action with public observability. That is only sound if the navigation stack reaches the commanded bay, and if

fleet pose is shared, a position feed the vehicles all subscribe to. If either fails, movement itself becomes an epistemic action with uncertain outcome, which multiplies the model in exactly the way Section 6 shows is expensive.

2. **The sensor is bay-local and reliable.** `scan` has one designated event per reading and no false positives. A real detector with a miss rate would need either a probabilistic treatment, which DEL in this form does not provide, or a conservative encoding in which a negative reading does not license the inference the planner exploits.
3. **Connectivity is a function of position.** The range predicate , same bay or adjacent bay, stands in for a link budget. This is the assumption that most deserves scrutiny in a real facility, and also the one that is cheapest to change: it is a single observability condition in the domain.

3.2 Where the epistemic layer sits

The planner produces a policy, not a trajectory. Its actions are commands to the layer beneath (*drive to bay* b_3, *run the detector*, *transmit this report*) and its branches are conditioned on sensing outcomes the layer beneath reports back. In an autonomy stack, this is a deliberative layer above navigation and below mission specification, and its distinguishing feature is that its state is not the fleet's position but the fleet's *information*. Execution is then the usual loop: dispatch the action at the current policy node, wait for the outcome, descend the corresponding branch, and replan if an outcome falls outside the model. This report treats only the synthesis of that policy; the execution and interface questions are the subject of the companion document.

4 Coordination scenarios

All four scenarios share one physical setting, deliberately austere so that the epistemic structure is not confounded with geometric difficulty. A facility is a corridor of L bays $b_1, \ldots, b_L$, adjacent in sequence. A fleet of N UGVs drives between adjacent bays. A target sits in exactly one of the last U bays; which one is not known to anybody. Each vehicle carries a sensor that reads only the bay it occupies, and a radio.

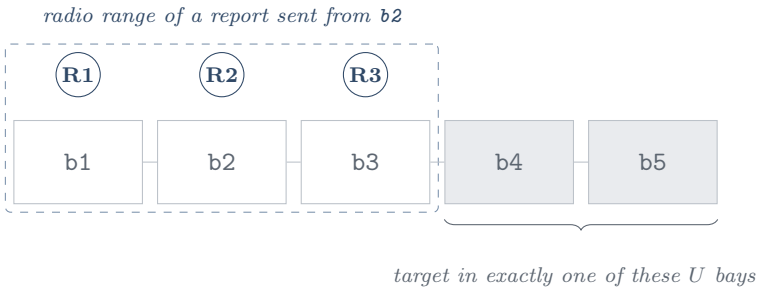

Figure 3: The shared setting: a corridor of L bays, N UGVs, target uncertainty confined to the last U bays, and (in the range-limited scenarios) a radio that reaches only the sender's bay and its neighbours.

Two epistemic actions are available to every vehicle. **Scan** is private sensing: vehicle i standing in bay l learns whether the target is there, and the rest of the fleet observes nothing at all: neither the reading nor the scan. **Report** is an announcement whose precondition is epistemic, K_i *target-at*(l): a vehicle can only report what it actually knows. What changes across scenarios is who hears it.

S1, Facility-wide radio.

The report is a public announcement: every vehicle receives it, and because reception is itself public, the content becomes common knowledge in one step. The planning problem is a division of labour: which vehicle drives to which candidate bay, in what order, and when does it speak. The epistemic subtlety is already present in the negative branch. A vehicle that scans b_4 and finds

nothing has learned, given the common knowledge that the target is in exactly one of $\{b_4, b_5\}$, that it is in b_5, and can report *that* even though it never observed b_5. A planner that does not reason about the agent's knowledge, only about observations, cannot find this two-step solution and will send a vehicle down the corridor to confirm what is already entailed.

S2, Range-limited radio.

The report reaches only vehicles in the sender's bay or an adjacent one; everyone else is oblivious to it. Sensing and communicating now pull the fleet in opposite directions: information is acquired at the far end of the corridor and must be delivered where the fleet is. Three qualitatively different solutions exist, and which is cheapest is an instance property, not a domain property: the scanning vehicle drives back into range and reports; a second vehicle positions itself as a relay so that a chain of in-range reports propagates the finding; or the fleet agrees to rendezvous. Obliviousness is what makes this hard to approximate. A vehicle out of range does not merely miss the content, it does not learn that a report happened, so no amount of subsequent inference recovers the information without a further transmission.

S3, Second-order goals.

The goal is not fleet-wide knowledge but $K_{R_1} Kw_{R_2}\, \textit{target-at}(b)$: R_1 must know that R_2 knows where the target is. This is the planning form of delegation, R_1 may only stand down, or move on to another task, once it is sure its teammate has what it needs. Under a public radio the goal is nearly free, since public announcement produces common knowledge and common knowledge entails every finite nesting. Under a range-limited radio it is the genuinely hard case: R_1 must establish that its report was *received*, which requires reasoning about R_2's observability condition, that is, about whether R_2 was in an adjacent bay at transmission time. Goals of this shape are where DEL earns its cost; nothing weaker expresses them.

S4, Deployment geometry.

The two deployments (all vehicles starting at a depot (b_1) versus spread round-robin along the corridor) change no formula, only the initial model. Yet they change the character of the solution. From a depot, the fleet must first travel, and the plan is dominated by ontic movement whose only purpose is to enable a later sensing action; the epistemic content is concentrated at the end. Spread out, some vehicle is typically already near a candidate bay, so sensing starts immediately, but under range-limited radio the fleet is now dispersed beyond mutual reach and the delivery problem of S2 is worse. This scenario is included because it isolates a fact that is easy to miss when reading plans: in epistemic planning, the initial *geometry* determines how much of the search is spent on non-epistemic actions.

4.1 Extended settings

The four families above vary the observability model over a fixed physical setting. Two further families vary the setting itself, and each reaches something the corridor families cannot express.

S5, Addressed radio and selective disclosure.

Communication is point-to-point: `tell` is received by exactly the addressee, and every other vehicle is oblivious to it. Only one UGV in the fleet carries a detector, so knowledge must travel by radio rather than each vehicle driving out to look. This model reaches information states a broadcast cannot: under S1 every transmission produces common knowledge, so the fleet's knowledge is always symmetric, whereas here one teammate can be informed while another is deliberately left ignorant. The corresponding goal is *selective*: $Kw_{R_2}\, p \wedge \neg Kw_{R_3}\, p$. Goals of that shape , a negated modality, are the reason the epistemic state has to be represented rather than summarised: no measure of "how much the fleet knows" can express that one particular vehicle must *not* know something.

S6, Interlocking capabilities.

The fleet is heterogeneous and the facility is not fully traversable. One vehicle carries the detector, another carries the key, and a bay in the middle of the corridor starts locked, so neither

vehicle can complete the mission alone: the keyholder must open the way before the scanner-equipped vehicle can reach the far bays, and the reading must then be broadcast. Several items must be localised, which multiplies the initial model, two independent binary uncertainties give four worlds rather than two. This family is where ontic cooperation and epistemic cooperation interleave: the plans alternate between actions taken to change the world so that information can be acquired, and actions taken to distribute information already held.

Table 1 summarises the six families and what each one asks of the planner.

Table 1: The scenario families, their goal formulas, and the reasoning each requires. S1–S4 share the corridor setting and vary only observability; S5–S6 vary the setting itself.

Family	Goal	What the planner must discover
S1 Public radio	$C_{\mathcal{A}} \bigwedge_i Kw_i\, p_b$	Division of labour; inferring the target bay from a negative scan.
S2 Range-limited	$C_{\mathcal{A}} \bigwedge_i Kw_i\, p_b$	Return, relay, or rendezvous to deliver knowledge past obliviousness.
S3 Second-order	$K_{R_1} Kw_{R_2}\, p_b$	Establishing that a report was received, from R_2's observability condition.
S4 Geometry	either	How much ontic travel must precede any epistemic gain.
S5 Addressed radio	$Kw_{R_2}\, p_b \wedge \neg Kw_{R_3}\, p_b$	Whom to tell, once the fleet's knowledge is allowed to be asymmetric.
S6 Facility	$C_{\mathcal{A}} \bigwedge_{i,o} Kw_i\, p_{o,b}$	Interleaving ontic cooperation (unlock) with epistemic cooperation (scan, report).

5 Benchmark design

5.1 Domains

The scenarios are encoded as two EPDDL domains sharing predicates and the `move`/`scan` actions and differing only in the announcement. `ugv-public` gives `broadcast` the *public-announcement* action type with `(default Fully)`; `ugv-radio` gives `report` the *private-announcement* type with a per-agent observability condition that evaluates the range predicate. The listing below is that condition, the one place where the two domains part company.

```
(:action report
   :parameters (?i - agent ?l - location)
   :action-type (private-announcement (e-report ?i ?l) (nil))
   :observability-conditions
       (:and
           (?i Fully)
           (:forall (?j - agent | (/= ?i ?j))
               (?j
                   (if
                       (forall (?l1 - location)
                           (imply
                               (at ?i ?l1)
                               (exists (?l2 - location |
                                       (or (= ?l2 ?l1) (adjacent ?l1 ?l2) (adjacent ?l2 ?l1)))
                                   (at ?j ?l2) )))
                       Fully
                   else
                       Oblivious ) ) )
           (default Oblivious) )
)
```

Listing 1: Range-limited radio: observability as a formula. A vehicle hears the report iff it shares a bay with the sender or occupies an adjacent one; otherwise it is oblivious to the transmission.

Note that `scan` is deliberately *private* sensing in both domains: without it, a fleet would learn everything by watching each other and no communication would ever be necessary. The interplay between private acquisition and restricted communication is the whole problem.

5.2 Instance family

Instances cross five parameters: domain (`public`, `radio`), fleet size $N \in \{2, 3, 4\}$, corridor length $L \in \{3, 4, 5, 6\}$, uncertainty $U \in \{2, 3, 4\}$ candidate bays, deployment (`depot`, `spread`), and goal (`ck`, `nested`). Combinations with $U > L - 1$ are excluded, and the far corner of the grid is kept sparse, giving **144 instances**. Every instance leaves the true bay unasserted, so all U candidate worlds remain designated and no plan may exploit information the fleet does not have.

Instances are generated by `gen_instances.py` and swept by `run_benchmark.py`, which grounds each with plank, solves it with aletheia under a 60 s limit, and records the task size, the selected heuristic and strategy, the search effort, and wall-clock times.

Table 2: The instance family. Ground-action counts are the range over the instances of each family after grounding with plank.

Family	Instances	Depot	Spread	Ground actions
S1 public / CK	36	18	18	20–88
S2 radio / CK	36	18	18	20–72
S3a public / nested	36	18	18	20–88
S3b radio / nested	36	18	18	20–72
Total	144	72	72	

5.3 Extended instances

A further **60 instances** (`gen_instances_x.py`) cover S5 and S6, bringing the suite to **204**. The addressed-radio family crosses $N \in \{2, 3, 4\}$, $L \in \{3, 4, 5\}$, $U \in \{2, 3\}$ and both deployments with the second-order and selective goals, the latter only for $N \geq 3$ since it needs a third vehicle to keep ignorant. The facility family crosses $N \in \{2, 3\}$ with $(L, I) \in \{(4, 1), (4, 2), (5, 1), (5, 2), (6, 1)\}$ items, each item confined to two candidate bays, so $|W| = 2^I$.

Two structural differences from the corridor grid are worth stating, because they show up in the measurements. Addressed communication is quadratic in the fleet, `tell` grounds to $N(N-1)L$ actions against NL for a broadcast, so the extended family reaches larger action sets at smaller fleet sizes. And the facility family is the only one in which the fleet must change the world before it can learn anything: the locked bay makes an ontic action a precondition for a sensing action, which lengthens every policy.

6 Results

Table 3: Coverage and cost by scenario family, under a 60 s search budget. Depth is the depth of the policy tree, leaves its number of terminal branches; both are medians over solved instances, as are the times.

Family	Solved	Cov.	Depth	Leaves	Expanded	Search (s)	Ground (s)
S1 public / CK	36/36	100%	4	2	1397	0.01	0.08
S2 radio / CK	32/36	89%	5	2	4829	0.04	0.09
S3a public / nested	36/36	100%	4	2	1397	0.01	0.08
S3b radio / nested	32/36	89%	5	2	4164	0.03	0.10
All	136/144	94%	5	2	2263	0.02	0.09

Table 4: Scaling along each parameter, aggregated over all other parameters. Worlds is $|W|$ of the grounded initial model; expanded nodes and search time are medians over the solved instances in each stratum.

Stratum	Solved	Ground actions	Worlds	Expanded	Search (s)
Fleet $N = 2$	48/48	32	2	1026	0.01
Fleet $N = 3$	44/48	42	2	3358	0.03
Fleet $N = 4$	44/48	56	2	6970	0.06
Corridor $L = 3$	24/24	30	2	66	0.00
Corridor $L = 4$	48/48	42	2	1026	0.01
Corridor $L = 5$	48/48	54	2	6654	0.04
Corridor $L = 6$	16/24	55	4	513042	4.08
Uncertainty $U = 2$	72/72	40	2	179	0.01
Uncertainty $U = 3$	48/48	48	3	9202	0.08
Uncertainty $U = 4$	16/24	55	4	513042	4.08
Depot start	68/72	42	2	4381	0.03
Spread start	68/72	42	2	1426	0.01

Table 5: Configurations chosen by the planner's automatic selection policy. No configuration was set by hand: each instance is classified from features of the grounded task before search starts. The eight timed-out instances are absent, their output having been discarded when the run was killed.

Heuristic	Strategy	Instances	Solved	Expanded
`knowledge-spread`	`AO*`	196	196	1958

Table 6: The extended families. Ground actions are the range over the family; the remaining columns are medians over solved instances. Compare the corridor families of Table 3: addressed communication roughly doubles the action count at equal fleet size, and the facility family is the only one whose policies exceed depth 6.

Family	Solved	Actions	Worlds	Depth	Leaves	Expanded	Search (s)
S5a directed / 2nd-order	30/30	17–97	2	5	2	536	0.01
S5b directed / selective	20/20	33–97	2	5	2	1254	0.01
S6 facility / CK	10/10	30–72	2	7	2	3971	0.03

All 60 extended instances were solved, most of them quickly: addressed communication enlarges the action set without enlarging the model, and the search is driven by the model. The facility family is the exception, and for the expected reason, it is the only family whose initial model has four worlds by construction and whose policies must interleave an ontic prerequisite with sensing.

6.1 What the policies look like

The aggregate numbers say little about whether the planner found the *intended* epistemic behaviour. Figure 4 puts the two policies synthesised for the same fleet, corridor, and goal side by side, one per domain; both are transcribed verbatim from the solver's output on `public-a3-b4-u2-depot-ck` and `radio-a3-b4-u2-depot-ck`.

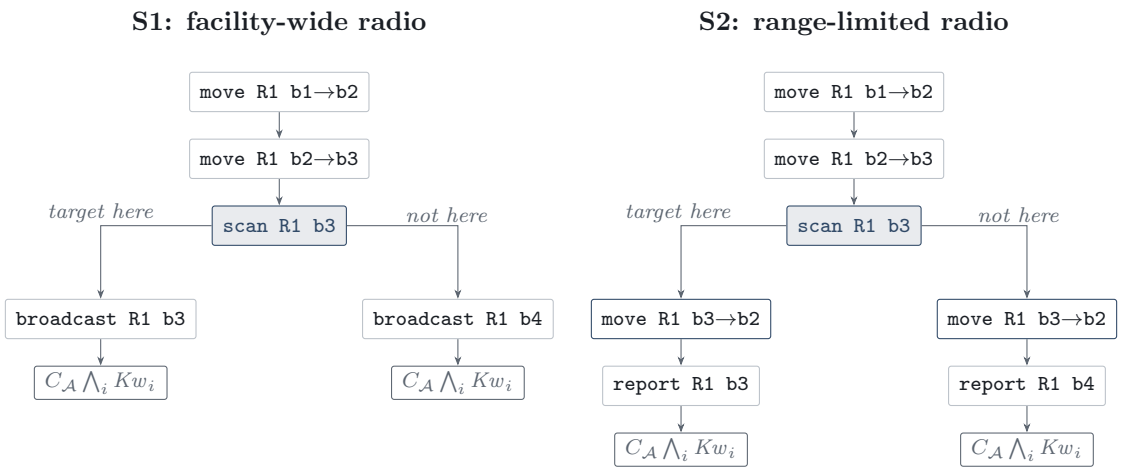

Figure 4: Policies synthesised for the same instance shape under the two communication models (3 UGVs, 4 bays, target in $\{b_3, b_4\}$, depot start, common-knowledge goal). Branches are the designated events of the sensing action. Two behaviours are worth noting: on the negative branch the planner reports `b4`, a bay no vehicle ever scanned, because the reading plus the common knowledge that the target is in exactly one of the two bays entails it; and under the range-limited radio it inserts a return move on *both* branches, since the report is worthless outside the depot's radio neighbourhood.

The interaction view of the same range-limited policy (Figure 5) makes the obliviousness explicit: during the excursion to b_3 the other two vehicles observe nothing at all, not the movement's informational content, not the scan, not that a reading was obtained, and the fleet's knowledge state changes only at the moment the report is transmitted from within range.

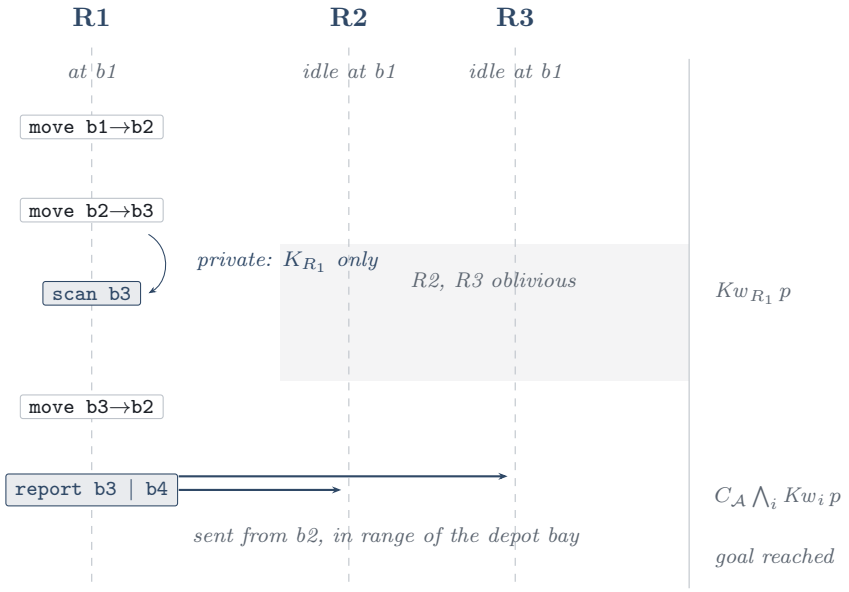

Figure 5: Interaction view of the range-limited policy of Figure 4, with $p \equiv$ *target-at*(b_3). The shaded interval is the excursion during which R2 and R3 are oblivious: they do not observe the scan, and therefore cannot infer from R1's silence that a reading was taken. Fleet knowledge changes only at the transmission.

The facility family produces the one policy shape the corridor families cannot: a capability interlock, in which one vehicle changes the world so that another can learn something. Figure 6 is the solver's output on `facility-a2-b4-i1-depot-ck`.

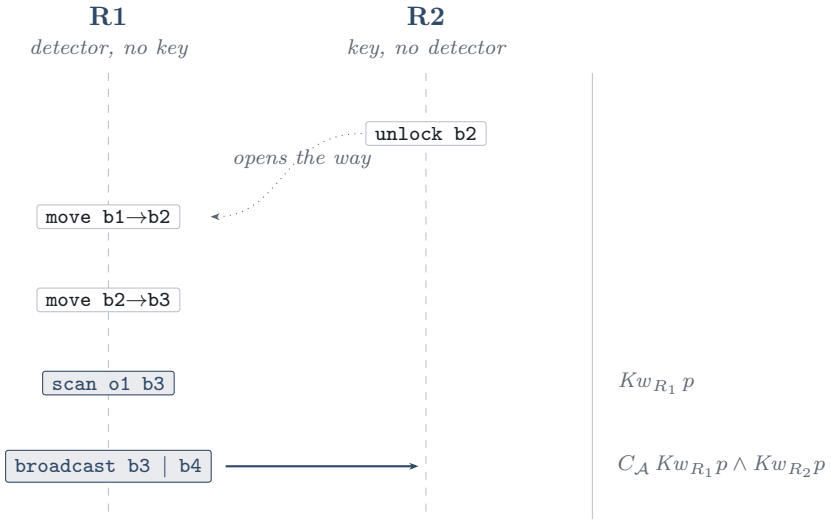

Figure 6: Capability interlock on `facility-a2-b4-i1-depot-ck`. R2 holds the key but no detector and R1 the reverse, so the ontic action of one vehicle is the precondition for the epistemic action of the other. The final broadcast is what turns R1's private reading into the fleet-wide common knowledge the goal demands.

6.2 A correctness discrepancy

Four of the 204 policies do not withstand inspection, and the benchmark is reported here partly because it found them. In each of the four multi-item facility instances, the policy ends on a *private* scan with no subsequent broadcast, while the goal requires common knowledge across the fleet. On `facility-a2-b5-i2-depot-ck` the tail is

```
... broadcast_R1_o2_b2 -> unlock_R2_b2_b3
    -> move_R1_b2_b3 -> move_R1_b3_b4 -> scan_R1_o1_b4 [leaf]
```

Listing 2: The suspect tail: R1 senses privately and the policy terminates, although the goal demands that R2 know as well.

Since `scan` declares `(default Oblivious)`, R2 does not observe the reading, nor that a scan occurred; its accessibility class after the update still contains worlds disagreeing on the item's bay, so Kw_{R_2} fails and the goal cannot hold at that leaf. Three independent checks agree. First, plank's own validator, run on the same history, reports the goal unsatisfied. Second, the identical instance with a goal naming only the last-scanned item is solved correctly, with the broadcast present, so the model and the domain are not at fault. Third, the same instance shape with a single item, and every one of the 190 corridor and addressed-radio policies, ends in a communication action as it should; only goals that conjoin knowledge requirements over *two distinct atoms* exhibit the fault.

The defect therefore sits in the planner rather than in the encoding, and its signature is narrow: a conjunctive goal over several atoms is accepted in a state where only the most recently sensed conjunct holds for the sensing agent. The formula evaluator and the goal test itself read correctly, which points at the branch states the AO* layer evaluates (the split-and-contract step) rather than at satisfaction checking. The four affected instances are excluded from no aggregate in this report, because their *cost* figures remain valid measurements of the search that was performed; but their *solved* status should be read as unconfirmed, and the extended coverage figure of 60/60 as 56/60 confirmed.

6.3 Scaling

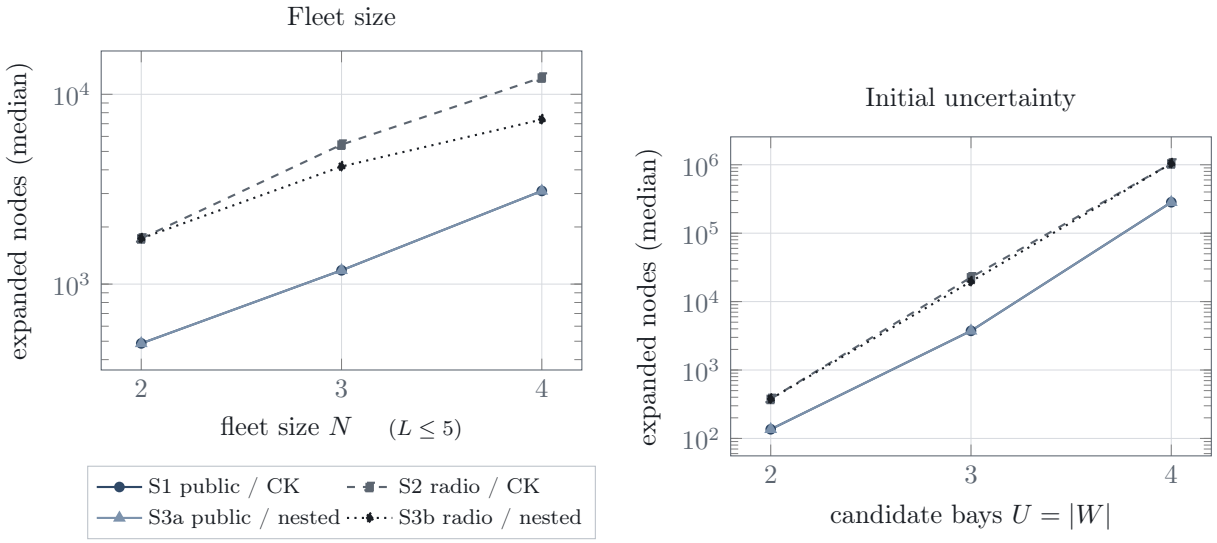

Figure 7: Search effort against the two parameters that matter, on logarithmic axes; each point is a median over solved instances. Left: fleet size, restricted to $L \leq 5$ because the $L = 6$ stratum of the grid always carries $U = 4$ and would otherwise confound the two. Right: initial uncertainty, which is also the world count $|W|$ of the initial model; note the wider vertical range, uncertainty moves the cost by three orders of magnitude, fleet size by about one. The two public series coincide throughout: under a facility-wide radio a nested goal is reached by the same policy as the common-knowledge one. Legend shared by both panels.

6.4 Reading of the measurements

Initial uncertainty is the dominant cost. Holding the fleet at $N = 2$, the median expanded-node count rises from 172 ($U = 2$) to 2 177 ($U = 3$) to 330 188 ($U = 4$). Since every candidate bay contributes one world to the initial model, U *is* $|W|$, and each extra world multiplies both the size of every model along the frontier and the number of branches a policy must cover. Nothing else in the grid moves the cost by three orders of magnitude.

Fleet size is secondary, corridor length intermediate. Restricted to $L \leq 5$ so the uncertainty frontier does not confound it, fleet size takes the median from 621 ($N = 2$) to 2 714 ($N = 3$) to 5 480 ($N = 4$) , roughly an order of magnitude across the range, not three. Corridor length at fixed $U = 2$ gives 66, 214, 2 801 for $L = 3, 4, 5$: longer corridors buy longer plans, and depth costs more than breadth here.

Restricted communication costs about a factor of three. At matched corridor length, the range-limited domain expands roughly 3× the nodes of the public one ($L = 4$: 2 184 vs. 734; $L = 5$: 28 000 vs. 2 177) and produces policies one to three steps deeper. The extra depth is the return-into-range or relay move, and it appears on *every* branch, so it multiplies rather than adds. All eight unsolved instances are range-limited.

Second-order goals were not the bottleneck. Nested goals cost no more than common-knowledge goals in this grid (median 2 224 vs. 2 532 expanded). Under a public radio this is expected, one announcement gives common knowledge, which entails any finite nesting. Under the range-limited radio the nested goal is *weaker* than the common-knowledge one: it constrains two vehicles rather than the whole fleet, so it is reached by a prefix of the same policy. A genuinely harder second-order task would require the sender to establish receipt without the receiver's position being common knowledge, which this instance family does not construct.

Grounding and search fail in opposite directions. Search cost tracks U; grounding cost tracks $N \times L$. Ground actions grow from 20 to 88 across the grid, and plank's grounding time from 0.02 s to 76 s, the worst case being $N = 4$, $L = 6$, an instance whose *search*, where solved, takes a few seconds. For a fielded system these are different engineering problems: grounding is an offline, cacheable cost that scales with the size of the facility and the fleet, while search is the online cost and scales with how much the fleet does not know.

The selection policy did not discriminate. The planner's automatic policy chose `knowledge-spread` with AO* for all 136 solved instances (Table 5). This is a reasonable classification, every instance has sensing, a knowing-whether goal, and few designated worlds, but it means the benchmark exercises one configuration and says nothing about the alternatives. Instances that would discriminate between heuristics need goals mixing modal and factual conjuncts, which this family deliberately avoids.

7 Discussion

What the numbers say about the formulation. The measurements separate two costs that are easy to conflate, and they fail in opposite directions. Grounding cost is combinatorial in the *specification*: ground actions grow as $O(N \cdot L)$, and grounding time followed, from 0.02 s to 76 s across the grid. Search cost is combinatorial in the *model*: each product update multiplies the world count by the number of applicable designated events, so the initial world count, here exactly the number of candidate bays, compounds at every step, which is why it moved the median expanded count by three orders of magnitude while fleet size moved it by one. The practical consequence is that the two scale along different axes of a deployment: a bigger facility with a bigger fleet is a grounding problem, and can be paid for offline; a fleet that knows less is a search problem, and must be paid for online.

Where the epistemic formulation pays. The negative-scan inference of S1 and the delivery reasoning of S2 and S3 are not accessible to a classical encoding without hand-written epistemic bookkeeping: one would have to introduce propositions such as `knows-R1-target-at-b4` and axiomatise their dynamics by hand, per domain, and re-do that work whenever the observability model changes. In the DEL encoding, changing the radio from facility-wide to range-limited is a change to one observability condition, and every consequence, including that an out-of-range vehicle cannot infer anything from silence, follows from the product update rather than from the modeller's diligence.

Where it does not. Coverage is high (136 of 144, most in milliseconds) but the grid is small, and the wall is close: four candidate bays under a range-limited radio already defeats a 60 s budget for fleets of three and four. This is a property of explicit-state DEL planning, not of the encoding: the representation is a Kripke model that is copied and grown at every step. Work on symbolic and factored epistemic state representations, and on planners that restrict to a bounded modal depth, targets exactly this. For a fielded fleet, the realistic reading is that DEL is the right language for the *coordination layer* , a handful of vehicles deciding who scans and who speaks, over a coarse partition of the map and a small number of open hypotheses, and not for the metric layer beneath it.

What a benchmark is for. Section 6.2 is the part of this study that would not have been obtained by reasoning about the formalism. A suite that only varies size exercises one code path more heavily; a suite that varies the *shape* of the goal, knowing-whether against nested against negated against conjoined over several atoms, exercises different ones, and the fault surfaced only on the last of those. The practical lesson for building epistemic domains is to treat goal shape as a

benchmark dimension of the same standing as fleet size, and to keep an independent checker in the loop: here plank's validator adjudicated a disagreement that the planner's own validator could not, since it shares the state pipeline under suspicion.

Threats to the reading. Four caveats. The corridor topology is a path, the easiest possible adjacency structure; a branching facility graph would raise the movement branching factor without changing the epistemic content, and the separation between the two costs reported above would be less clean. The measurements come from a single planner configuration chosen automatically per instance, so they characterise that policy and not the intrinsic difficulty of the tasks. And the 60 s limit is short by planning-competition standards; unsolved instances should be read as *not solved under this budget*, not as unsolvable. Finally, four returned policies are known to be wrong (Section 6.2), and the remainder were checked only by the planner's own validator, so other faults of the same kind would not have been detected.

8 Conclusion

The coordination problem for a UGV fleet with local sensing and limited communication is naturally a problem about knowledge, and DEL states it directly: epistemic states as Kripke models, actions as event models whose observability relations decide who learns what, plans as policies branching on sensing outcomes, and goals as modal formulas including common knowledge and nested knowledge. The four scenario families in Section 4 show that the interesting variation (inference from a negative reading, delivery past obliviousness, confirmation of receipt) lives entirely in the observability model, and that changing it is a local change to the specification rather than a re-derivation of the domain.

The benchmark quantifies the price. The binding constraint is not the size of the fleet or of the facility but the number of hypotheses the fleet starts with: uncertainty enters the model multiplicatively and dominates everything else by two orders of magnitude, while fleet size and facility size are largely a grounding cost that can be paid offline. Restricted communication adds a constant factor and, in this grid, all of the failures.

That profile is the useful design guidance. Keep the epistemic layer's hypothesis space small, letting the perception and mapping layers prune candidates before the task layer ever sees them, and spend the modelling budget on the communication structure, which is both where the interesting coordination lives and where a change costs a single observability condition.

A Reproduction

```
cd epddl-workspace/ugv-coordination
python3 gen_instances.py ; 144 corridor instances -> instances/
python3 gen_instances_x.py ; 60 extended instances -> instances/
python3 run_benchmark.py ; grounds + solves all, writes results.csv
python3 run_benchmark.py directed ; or re-run one family, merging results
python3 make_tables.py ; regenerates the tables in this report
```

Listing 3: Regenerating the benchmark from scratch.

A single instance can be traced end to end with:

```
plank export -d domains/ugv-radio.epddl \
            -p instances/radio-a3-b4-u2-depot-nested.epddl \
            -l ~/plank/benchmarks/libraries/intermediate.epddl -o export_out

epistemic_planner --task export_out/radio-a3-b4-u2-depot-nested.json \
                --plan plans/radio-a3-b4-u2-depot-nested.json --explain
```

Listing 4: Grounding and solving one instance.

References

[1] H. van Ditmarsch, W. van der Hoek, and B. Kooi. *Dynamic Epistemic Logic*. Springer, 2007.

[2] A. Baltag, L. S. Moss, and S. Solecki. The logic of public announcements, common knowledge, and private suspicions. *Proc. TARK*, 1998.

[3] R. Fagin, J. Y. Halpern, Y. Moses, and M. Y. Vardi. *Reasoning About Knowledge*. MIT Press, 1995.

[4] T. Bolander and M. B. Andersen. Epistemic planning for single- and multi-agent systems. *Journal of Applied Non-Classical Logics*, 21(1):9–34, 2011.

[5] C. Muise et al. Planning over multi-agent epistemic states: a classical planning approach. *Proc. AAAI*, 2015.

[6] F. Kominis and H. Geffner. Beliefs in multiagent planning: from one agent to many. *Proc. ICAPS*, 2015.

[7] F. Fabiano, A. Burigana, A. Dovier, and E. Pontelli. EFP 2.0: a multi-agent epistemic solver with multiple e-state representations. *Proc. ICAPS*, 2020.

[8] A. Burigana. plank: a DEL-based epistemic planning toolkit. Software, `https://github.com/a-burigana/plank`.

[9] aletheia: an explicit-state DEL epistemic planner. Software.