DEPTH: A DEL-BASED EPISTEMIC PLANNER WITH TIER HEURISTICS

Halim Djerroud

Université Paris-Saclay, UVSQ, LISV, 78140, Vélizy-Villacoublay, France
halim.djerroud@uvsq.fr

ABSTRACT

We present DEPTH (DEL Epistemic Planner with Tier Heuristics), our entry to the IPC 2026 Epistemic Planning Track, built on top of the plank toolkit. Its core algorithm, Epistemic H*, is a two-queue best-first search adapted to Dynamic Epistemic Logic (DEL): a primary queue advances toward the goal, while a secondary queue retains nodes that fail to make progress but may correspond to *setup actions*—announcements or sensing steps that enable later goal-achieving actions. When the primary queue is empty, the secondary queue is flushed, guaranteeing completeness. Goal progress is estimated by a deliberately coarse tier heuristic: the number of unsatisfied subformulas of the decomposed goal. Bisimulation contraction is applied at every step. We prove completeness for finite DEL tasks. Preliminary results show that DEPTH significantly outperforms plain BFS on problems with large Kripke models.

Index Terms— epistemic planning, dynamic epistemic logic, heuristic search, bisimulation, IPC 2026

1. INTRODUCTION

Epistemic planning extends classical planning with the ability to reason about the knowledge and beliefs of agents. It arises naturally in multi-agent coordination, human-robot interaction, and security protocol verification [1], where the success of a plan depends not only on the state of the world but also on what agents know or believe about it.

Dynamic Epistemic Logic (DEL) [2, 3] provides a principled semantic framework for this setting. Epistemic states are represented as Kripke models [4], and actions are modelled as event models whose execution updates the state via the *product update* operator. Despite its expressive power and solid theoretical foundations, DEL-based planning has seen limited adoption in practice. A key obstacle is the state-space explosion that follows from the product update: each action application can multiply the number of possible worlds, making blind search strategies such as BFS quickly intractable.

A further obstacle has been the fragmentation of the field: existing epistemic planners target different fragments of DEL, rely on ad hoc input languages, and are rarely evaluated on common benchmarks, making systematic comparison difficult. The first Epistemic Planning Track of the International Planning Competition (IPC 2026) [5] directly addresses this issue by introducing EPDDL [6], a unified language covering a wide range of DEL fragments, together with the plank toolkit [7] for parsing, grounding, and validation.

In this paper we present **DEPTH** (DEL Epistemic Planner with Tier Heuristics), our entry to IPC 2026. DEPTH goes beyond the BFS baseline provided by plank by introducing a heuristic search strategy called *Epistemic H**. The algorithm is a transposition to the epistemic setting of the Head Star (H*) motion planning algorithm originally introduced in [8] for Navigation Among Movable Obstacles (NAMO). Its core idea is to decompose the goal formula into a set of independent subgoals and to use the number of unsatisfied subgoals — the *tier value* — as the heuristic guiding a two-queue best-first search. Bisimulation contraction is applied at every step to keep the state space compact. We prove that the algorithm is complete for finite DEL planning tasks.

The design objective of DEPTH is not plan-length optimality, but scalable epistemic reachability under the full DEL semantics — finding *a* plan rather than the shortest, while keeping the product-update semantics intact. This framing should guide the reading of the algorithmic choices and the experimental evaluation that follow.

Contributions.

- Epistemic H*: a two-queue best-first search for DEL-based epistemic planning, designed to preserve *setup actions* that pure greedy search would discard.
- A goal decomposition procedure for multi-agent modal formulas, used by a deliberately coarse but lightweight tier heuristic.
- A completeness proof for Epistemic H* on finite DEL tasks.
- Preliminary experimental results on the IPC 2026 sample benchmarks.

The rest of the paper is organised as follows. Section 2 recalls the DEL framework and the original H* algorithm. Section 3 presents Epistemic H* and its completeness proof. Section 4 reports preliminary experimental results. Section 5 discusses related work. Section 6 concludes.

2. BACKGROUND

2.1. Dynamic Epistemic Logic

A *Kripke model* (epistemic state) is a tuple $M = (W, R, L, W^*)$ where W is a finite set of possible worlds, $R : Ag \to 2^{W \times W}$ assigns an accessibility relation to each agent, $L : W \to 2^{At}$ labels worlds with true atoms, and $W^* \subseteq W$ is the non-empty set of designated worlds representing the actual situation.

An *action model* (epistemic action) is a tuple $A = (E, Q, pre, post, E^*)$ where E is a set of events, $Q : Ag \to 2^{E \times E}$ gives observability relations, $pre : E \to \mathcal{L}$ assigns preconditions, $post : E \to (At \rightharpoonup \mathcal{L})$ assigns to each event a partial map from atoms to postcondition formulas, and $E^* \subseteq E$ is the set of designated events. Executing A in M is performed via the *product update* [2, 3], which produces a new Kripke model $M \otimes A$ whose domain consists of pairs $(w, e) \in W \times E$ such that $M, w \models pre(e)$.

A *DEL planning task* is a triple $(M_0, \mathcal{A}, \gamma)$ where M_0 is the initial epistemic state, $\mathcal{A}$ is a finite set of action models, and γ is a goal formula. A *plan* is a sequence of actions $[a_1, \ldots, a_n] \in \mathcal{A}^*$ such that the state obtained by successive product updates satisfies γ.

2.2. Bisimulation Contraction

Two Kripke models are *bisimilar* if there exists a relation between their worlds that preserves truth of all modal formulas in both directions. Bisimilar states are logically indistinguishable: no formula of the epistemic language can tell them apart [9].

The *bisimulation contraction* of a state M is the unique (up to isomorphism) minimal Kripke model bisimilar to M. It is obtained by quotienting M by its largest bisimulation. Two states with the same contraction are logically equivalent and can be identified during search, which is the key mechanism DEPTH uses to keep the frontier compact.

2.3. The Original H* Algorithm

Epistemic H* reuses the structural pattern introduced by the H* motion planning algorithm of [8]: two priority queues OPEN and INCONS, and a flush mechanism that moves the contents of INCONS back to OPEN whenever the latter becomes empty. The robotics-specific aspects of the original algorithm (spatial heuristic, grid representation, reactive navigation) are not relevant in our setting; only the queue discipline transfers. In the epistemic adaptation, OPEN holds nodes whose tier strictly improves over their parent, INCONS holds the rest, and the flush mechanism is what underwrites completeness despite an inadmissible heuristic, as shown in Section 3.6.

2.4. EPDDL and plank

EPDDL [6] is the input language of IPC 2026. It covers three increasing levels of expressivity: *Basic* (public actions, $S5_n$ frames), *Intermediate* (private actions, common knowledge, $KD45_n$ frames), and *Hard* (arbitrary DEL actions and frames). Action types are defined in reusable libraries, allowing benchmarks to target specific DEL fragments without modifying domain or problem files. DEPTH supports all three levels, as the DEL semantics is handled entirely by plank.

The plank toolkit [7] parses and grounds EPDDL specifications into a JSON representation consumed by DEPTH, and also provides a plan validator used to verify the correctness of solutions.

3. THE DEPTH PLANNER

3.1. Overview

DEPTH reads a ground JSON epistemic planning task produced by plank and searches for a plan using the *Epistemic H** algorithm. At each step, the successor state is reduced to its bisimulation contraction before being inserted into the frontier, so that two states that are logically indistinguishable are always identified as identical regardless of how they were reached.

3.2. Goal Decomposition

A key ingredient of Epistemic H* is the decomposition of the goal formula γ into a set of *facts* $F = \{f_1, \ldots, f_k\}$ that can be checked independently. The decomposition proceeds as follows:

1. **Conjunction flattening.** If $\gamma = \varphi_1 \wedge \cdots \wedge \varphi_n$, the procedure is applied recursively to each conjunct. This is sound because $s \models \gamma$ iff $s \models \varphi_i$ for all i.
2. **Multi-agent modal splitting.** Let $G \subseteq Ag$ be a group of agents. A modal formula $\mathrm{Kw}_G\, \psi$ ("every agent in G knows whether ψ") with $|G| > 1$ is split into $|G|$ singleton formulas $\mathrm{Kw}_{\{a\}}\, \psi$, one per $a \in G$. The same applies to the group box $[\cdot]$, its dual $\langle \cdot \rangle$, common knowledge C, and its existential dual. Splitting is conservative: if the original formula holds then all singletons hold, but not necessarily vice versa. The split therefore yields an *over-approximation* of the goal, which is acceptable for an inadmissible heuristic.
3. **Atomic and modal base cases.** Propositional atoms, negations, disjunctions, and singleton modal formulas are kept as-is.

If the decomposition yields the empty set (e.g. $\gamma = \top$), γ itself is used as the sole fact. The resulting set F contains $k \geq 1$ independent subgoals.

3.3. Tier Heuristic

Given the fact set F, the *tier value* of a state s is:

$$h(s) \;=\; \big|\{\, f \in F \mid s \not\models f \,\}\big|$$

that is, the number of facts not yet satisfied in s. By construction, $s \models \gamma$ implies $h(s) = 0$; the converse holds for the modal fragments where the decomposition is exact, but not in general, since splitting group modalities yields an over-approximation (see Section 3.7). The tier heuristic is *inadmissible* in general: applying a single action may simultaneously satisfy several facts (causing the tier value to drop by more than one), or may satisfy none. It is nonetheless informative because epistemic actions tend to propagate knowledge incrementally, making the number of unsatisfied knowledge subgoals a meaningful proxy for distance to the goal.

This design departs deliberately from A^*. In the epistemic setting, actions carry no natural cost: one product update step has the same logical weight regardless of its content, making plan-length optimality secondary to finding *any* plan at all. The g-component of A^* would add bookkeeping without exploiting the epistemic structure. Moreover, computing a truly admissible heuristic in DEL would require solving sub-problems of complexity at least as high as the original task [1]. The tier heuristic is deliberately coarse, but it is computationally lightweight (a single model-checking call per fact) and naturally aligned with the conjunctive structure of epistemic goals.

3.4. Epistemic H* Search

Pure greedy best-first search (GBFS) is fragile in the epistemic setting because many actions are *setup actions* — they do not directly satisfy any goal subformula but enable future actions that do. For example, an announcement that makes an agent aware of another agent's private channel does not immediately satisfy any knowledge goal, yet it may be the only way to enable a subsequent sensing action that does. A pure greedy strategy would deprioritise or permanently skip such nodes. The two-queue structure solves this: a node that fails to decrease the tier goes to INCONS rather than being discarded, and is reconsidered when OPEN is exhausted.

Algorithm 1 gives the pseudocode of Epistemic H*. The search maintains two priority queues, both ordered by tier h with node depth as tie-breaker (smaller depth first): OPEN collects nodes that strictly improve the tier of their parent, while INCONS collects the rest. When OPEN is exhausted, all nodes from INCONS are flushed into OPEN and the search resumes. A global visited set V (keyed on the canonical bisimulation contraction) prevents any state from being expanded more than once.

Algorithm 1 Epistemic H*

```
Require: planning task $(s_0, \mathcal{A}, \gamma)$; fact set $F = \mathsf{decompose}(\gamma)$
Ensure: plan $\pi$ or UNSOLVABLE
if $s_0 \models \gamma$ then return $\langle\rangle$
end if
$s_0 \leftarrow \mathsf{contract}(s_0)$
$V \leftarrow \{s_0\}$; OPEN $\leftarrow \{(s_0, h(s_0), 0, \bot, \bot)\}$; INCONS $\leftarrow \emptyset$
while OPEN $\neq \emptyset$ or INCONS $\neq \emptyset$ do
    if OPEN $= \emptyset$ then
        move all nodes from INCONS to OPEN
    end if
    $n \leftarrow$ POPMIN(OPEN)
    for each action $a \in \mathcal{A}$ do
        if $\neg\,\mathsf{applicable}(n.s,\ a)$ then continue
        end if
        $s' \leftarrow \mathsf{contract}(\mathsf{update}(n.s,\ a))$
        if $s' \models \gamma$ then return EXTRACTPLAN($n$) $+[a]$
        end if
        if $s' \notin V$ then
            $V \leftarrow V \cup \{s'\}$
            $n' \leftarrow (s',\ h(s'),\ n.depth + 1,\ a,\ n)$
            if $h(s') < h(n.s)$ then
                OPEN.push($n'$)
            else
                INCONS.push($n'$)
            end if
        end if
    end for
end while
return UNSOLVABLE
```

3.5. Illustrative Example

Consider a three-agent task with atoms $\{p, q, r\}$, agents $\{A, B, C\}$, and goal $\gamma = K_A\, p \wedge K_B\, q \wedge K_C\, r$ (each agent should know its designated fact). Goal decomposition yields three independent facts: $F = \{K_A\, p,\ K_B\, q,\ K_C\, r\}$.

In the initial state s_0 no agent has yet acquired any knowledge, so $h(s_0) = 3$. Figure 1 illustrates the resulting search tree. The search proceeds as follows:

1. Action a_1 (agent A performs a sensing action and learns p): the successor s_1 satisfies $K_A\, p$ but not the other two facts, so $h(s_1) = 2 < h(s_0) = 3$. Node s_1 goes to OPEN.
2. Action a_2 (agent B learns q) applied to s_1: $h(s_2) = 1 < h(s_1)$. Node s_2 goes to OPEN.
3. Action a_3 (agent C learns r) applied to s_2: $s_3 \models \gamma$. The algorithm returns the plan $[a_1, a_2, a_3]$.

At each step the tier strictly decreases, so all successors go directly to OPEN and the search is purely greedy. If an action

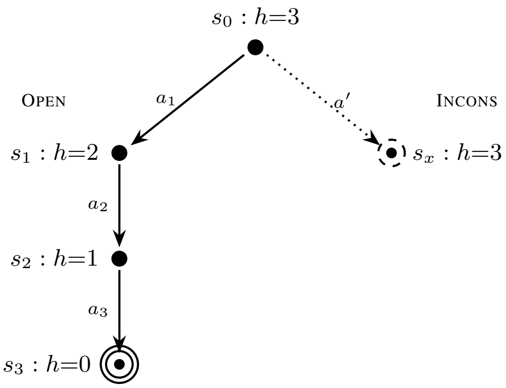

Fig. 1. Search tree for the illustrative example. Solid edges lead to OPEN (tier strictly decreases); the dotted edge leads to INCONS (no tier improvement). The goal state s_3 is shown with a double circle.

had failed to decrease the tier (e.g. an irrelevant announcement), the resulting node would have gone to INCONS and been deferred until OPEN was exhausted, avoiding wasteful exploration of unpromising branches.

3.6. Completeness

The tier heuristic is inadmissible, so Epistemic H* may return a plan that is not shortest. We now show that despite this, the algorithm never misses an existing solution.

The proof exploits two structural properties: (i) the flush mechanism guarantees that no generated node stays in INCONS forever, and (ii) the goal test precedes the visited check.

Lemma 1 (Finite reachability)**.** *The set of states reachable from s_0 via $\mathcal{A}$, up to bisimulation, is finite.*

Proof. The bisimulation contraction maps every Kripke model to a canonical minimal representative. Over a finite set of atoms At and agents Ag, the number of non-bisimilar pointed Kripke models is finite [9], so the reachable set is finite. □

Lemma 2 (Every reachable state is eventually expanded)**.** *If a state s is reachable from s_0 and the algorithm has not returned, then s will eventually be removed from* OPEN *and expanded.*

Proof. By induction on the length of a shortest path $s_0 \xrightarrow{a_1} s_1 \xrightarrow{a_2} \cdots \xrightarrow{a_n} s_n = s$.
Base case ($n = 0$). s_0 is inserted into OPEN at initialisation and will be expanded.
Inductive step. Assume s_{i-1} is eventually expanded. During that expansion, action a_i is tried (a_i is applicable in s_{i-1} by definition of the reachability path). If $s_i \models \gamma$ the algorithm returns (not our case by hypothesis). If $s_i \in V$, it was previously generated and is either in OPEN, in INCONS, or already expanded; in all three cases the conclusion holds. Otherwise $s_i \notin V$ and it is inserted into OPEN or INCONS.

In both cases s_i will be expanded: nodes in OPEN are dequeued directly; nodes in INCONS are moved to OPEN whenever OPEN becomes empty, which must happen in finite time because the state space is finite (Lemma 1) and each state enters V at most once. □

Theorem 1 (Completeness)**.** *If there exists a plan for the task, Epistemic H* returns a plan.*

Proof. Let $\pi = [a_1, \ldots, a_n]$ be a plan, and let s_n be the goal state it reaches. By Lemma 2, s_{n-1} is eventually expanded. During that expansion, $s_n = \mathsf{contract}(\mathsf{update}(s_{n-1}, a_n))$ is generated and the condition $s_n \models \gamma$ is checked at line 14 of Algorithm 1, *before* the visited check at line 16. Therefore the algorithm returns a plan. □

Theorem 1 guarantees that a plan is found whenever one exists, but not that it is shortest.

3.7. Limitations and Design Tradeoffs

DEPTH is a first-iteration planner submitted to the inaugural IPC 2026 epistemic track, and we deliberately favoured a simple, transparent design over a more aggressive one. Several limitations follow from this choice and are worth stating explicitly.

Heuristic granularity. The tier value counts the number of unsatisfied facts but is blind to their internal structure: a goal $K_A(p \wedge q \wedge r \wedge s)$ contributes the same unit as $K_A\, p$, even though the former is harder to achieve. Two states with the same tier value may therefore be very different in practice. A finer tier heuristic — e.g. one that recurses into the structure of each fact — is an obvious direction for future work, but was kept out of the competition submission to keep the algorithm easy to audit.

Goal decomposition is conservative. Splitting a multi-agent modal goal into per-agent singletons (Section 3.2) yields an over-approximation: a state satisfying every singleton fact does not necessarily satisfy the original goal. This is particularly visible for common knowledge: $\mathrm{C}_{\{A,B\}}\, p$ split into $K_A\, p \wedge K_B\, p$ can lead the tier heuristic to report $h(s) = 0$ while the actual goal still fails. Correctness is unaffected because the goal test uses the original γ, but the tier heuristic can mis-rank states near common-knowledge goals.

Inadmissibility cost. As shown in Section 4 on Coin-in-the-Box p3 and p5, Epistemic H* can return plans one step longer than BFS. The two-queue mechanism mitigates the worst cases (setup actions are preserved through INCONS) but does not recover optimality. A version with an admissible heuristic — or post-hoc plan shortening — would be a natural follow-up.

Cost of bisimulation contraction. Contraction is applied to every successor before insertion into the frontier. On states with many worlds and agents, this can dominate the per-node cost. Empirically it remains the right tradeoff because the frontier stays compact, but a lazy or incremental variant could pay off on harder instances.

4. PRELIMINARY EXPERIMENTS

We report preliminary results on the sample benchmark problems made available by the IPC 2026 organisers prior to the competition. These instances are intended as smoke-test examples; the full competition benchmark set, as well as a comparison with other competing planners, will only be available after the competition concludes. A comprehensive experimental evaluation will be presented in the extended version of this paper.

Experiments were run with a 60-second timeout on a standard desktop machine. Table 1 reports plan length and runtime for Epistemic H* and plain BFS (used here as a baseline).

Table 1. Preliminary results on IPC 2026 sample problems (60 s timeout).

Domain	$\|W\|$	BFS	Epistemic H*
Coin-in-the-Box p1	2	2 steps, 0 ms	2 steps, 0 ms
Coin-in-the-Box p2	2	4 steps, 1 ms	4 steps, 1 ms
Coin-in-the-Box p3	2	5 steps, 3 ms	6 steps, 2 ms
Coin-in-the-Box p4	2	6 steps, 5 ms	6 steps, 3 ms
Coin-in-the-Box p5	2	5 steps, 3 ms	6 steps, 3 ms
Grapevine p1	32	timeout	5 steps, 15 ms
Active-Muddy-Child	32	no plan, 4 ms	no plan, 4 ms
Blocks-World p1	1	no plan, 0 ms	no plan, 0 ms
Consecutive-Numbers	5	no plan, 0 ms	no plan, 0 ms
Gossip p1	32	timeout	timeout

The heuristic guidance of Epistemic H* is decisive on Grapevine p1 (32 worlds, 35 actions), where BFS exhausts the time limit while H* finds a 5-step plan in 15 ms. On small problems (Coin-in-the-Box, 2 worlds), both algorithms perform comparably — the search space is too small for the tier heuristic to provide a significant advantage.

On Coin-in-the-Box p3 and p5, Epistemic H* returns plans that are one step longer than those found by BFS. Inspection of the plans explains the gap. The BFS plan establishes communication channels with both target agents first (`signal_A_B`, `signal_A_C`) and then issues a single `shout-tails_A` which reduces the tier value from 2 to 0 in one step. Epistemic H*, by contrast, issues a `shout-tails_A` as soon as the first signal is in place: that step strictly decreases the tier value (2→1) and is therefore prioritised by the tier heuristic, but it forces a second `shout-tails_A` after the second signal. This is the canonical inadmissibility cost: a local tier improvement is incompatible with a globally shorter plan. The plan returned by Epistemic H* is still correct—we verified all plans reported in Table 1 using the plank validator. BFS, being exhaustive, always finds a shortest plan but at a higher computational cost (and times out on Grapevine).

The domains returning "no plan" (Active-Muddy-Child, Blocks-World, Consecutive-Numbers) are genuinely unsolvable at the given instance level, as confirmed by the plank reference BFS. Gossip p1 (32 worlds, 20 actions) remains unsolved by both algorithms within 60 s, pointing to the need for stronger heuristics in this domain.

5. RELATED WORK

Several epistemic planners have been proposed in the literature, targeting different fragments of DEL or equivalent formalisms. We organise them along three axes that situate DEPTH's contribution.

Foundational DEL planning. Bolander and Andersen [1] introduced the modern DEL-based formulation of epistemic planning, defining tasks as triples $(s_0, \mathcal{A}, \gamma)$ over pointed Kripke models with event-model actions. They established the formal landscape (decidability boundaries, basic algorithms) but did not propose a heuristic search procedure. DEPTH inherits this formulation directly and supplies the heuristic component their setting leaves open.

Compilation-based approaches. Muise et al. [10] compile multi-agent epistemic planning into classical planning by enumerating possible worlds and encoding knowledge as classical fluents over a finite belief-base representation. The approach is attractive because it inherits the mature heuristic machinery of classical planners (FF, FastDownward), but pays for this in two ways. First, it does not generalise to large or dynamically generated Kripke models, where the fluent encoding explodes. Second, it loses access to the product-update semantics of DEL: actions must be expressible as classical operators, which restricts the supported DEL fragment. DEPTH chose the opposite tradeoff — keeping the full DEL semantics through plank and developing a heuristic that operates directly on Kripke models.

Direct DEL-based planners. Engesser et al. [11] propose a DEL-based planner for cooperative multi-agent settings with implicit coordination, focusing on the question of when agents should act based on their individual epistemic states. Fabiano et al. [12] develop EFP and PG-EFP, epistemic forward planners supporting multi-agent domains with partial observability; they use possibility structures rather than full Kripke models, which keeps state sizes manageable but constrains the supported language. Both approaches operate directly on epistemic states without compilation, like DEPTH, but neither employs a generic heuristic to guide the search — they rely on uninformed forward expansion or domain-specific control rules. The plank toolkit [7] includes a reference BFS planner that serves as our baseline; DEPTH extends this baseline with goal-decomposition guidance and a two-queue best-first mechanism.

Heuristic search in epistemic planning. To the best of our knowledge, DEPTH is the first DEL-based planner to apply a modal goal-decomposition heuristic combined with a deferred-queue search discipline. The idea of separating tier-improving from tier-neutral successors — rather than discarding the latter or treating them uniformly with the former — originates in the H* algorithm for navigation among movable obstacles [8]. In that robotics setting, the second queue stored states whose spatial heuristic did not strictly decrease but whose expansion was still required for completeness. The epistemic adaptation transfers the queue discipline, discarding the spatial heuristic and the reactive components, and replaces them with the tier heuristic of Section 3.3. We see this adaptation as a starting point: stronger epistemic heuristics could be plugged into the same two-queue framework with no algorithmic change.

6. CONCLUSION

We presented DEPTH, a DEL-based epistemic planner submitted to the first Epistemic Planning Track of IPC 2026. Its core algorithm, Epistemic H*, adapts the two-queue search structure of H* [8] to the epistemic setting, where its main role is to preserve *setup actions* that pure greedy search would discard. Goal progress is tracked by a deliberately coarse tier heuristic counting unsatisfied subformulas of the decomposed goal. Bisimulation contraction is applied at every step to keep the state space compact, and we proved that the algorithm is complete for finite DEL planning tasks despite the inadmissibility of its heuristic.

Preliminary results on the IPC 2026 sample benchmarks show that Epistemic H* significantly outperforms plain BFS on problems with large Kripke models (e.g. Grapevine, solved in 15 ms where BFS times out), while remaining competitive on smaller instances.

Future work. The main open challenge is scalability on domains such as Gossip, where both BFS and Epistemic H* time out. Several directions are worth exploring: (i) stronger heuristics that account for the structure of specific action types (e.g. sensing vs. announcement actions); (ii) an admissible variant of the tier heuristic to recover plan-length optimality; (iii) symmetry breaking and additional pruning based on bisimulation. A full experimental comparison with other IPC 2026 competitors will be presented in the extended version of this paper.

7. REFERENCES

[1] Thomas Bolander and Mikkel Birkegaard Andersen, “Epistemic planning for single and multi-agent systems,” *Journal of Applied Non-Classical Logics*, vol. 21, no. 1, pp. 9–34, 2011.

[2] Alexandru Baltag, Lawrence S Moss, and Sławomir Solecki, “The logic of public announcements, common knowledge, and private suspicions,” in *Readings in Formal Epistemology: Sourcebook*, pp. 773–812. Springer, 2016.

[3] Hans van Ditmarsch, Wiebe van der Hoek, and Barteld Kooi, *Dynamic Epistemic Logic*, vol. 337 of *Synthese Library*, Springer, Dordrecht, 2007.

[4] Ronald Fagin, Joseph Y. Halpern, Yoram Moses, and Moshe Y. Vardi, *Reasoning About Knowledge*, MIT Press, Cambridge, MA, 1995.

[5] Alessandro Burigana and Francesco Fabiano, “IPC 2026 epistemic planning track,” `https://sites.google.com/view/epistemic-competition/`, 2026.

[6] Alessandro Burigana and Francesco Fabiano, “The epistemic planning domain definition language: Official guideline,” *arXiv preprint arXiv:2601.20969*, 2026.

[7] Alessandro Burigana, “plank: A toolkit for DEL-based epistemic planning,” `https://github.com/a-burigana/plank`, 2026.

[8] Halim Djerroud, “Head star (H*): A motion planning algorithm for navigation among movable obstacles,” in *Proceedings of the 15th International Conference on Agents and Artificial Intelligence (ICAART)*. 2023, pp. 211–219, SCITEPRESS.

[9] Patrick Blackburn, Maarten de Rijke, and Yde Venema, *Modal Logic*, vol. 53 of *Cambridge Tracts in Theoretical Computer Science*, Cambridge University Press, 2001.

[10] Christian Muise, Vaishak Belle, Paolo Felli, Sheila A. McIlraith, Tim Miller, Adrian R. Pearce, and Liz Sonenberg, “Planning over multi-agent epistemic states: A classical planning approach,” in *Proceedings of the 29th AAAI Conference on Artificial Intelligence*. 2015, pp. 3327–3334, AAAI Press.

[11] Thorsten Engesser, Thomas Bolander, Robert Mattmüller, and Bernhard Nebel, “Cooperative epistemic multi-agent planning for implicit coordination,” in *Proceedings of the 9th Workshop on Methods for Modalities (M4M)*, 2017, vol. 243 of *Electronic Proceedings in Theoretical Computer Science*, pp. 75–90.

[12] Tiep Le, Francesco Fabiano, Tran Cao Son, and Enrico Pontelli, “EFP and PG-EFP: Epistemic forward search planners in multi-agent domains,” in *Proceedings of the 28th International Conference on Automated Planning and Scheduling (ICAPS)*. 2018, pp. 161–170, AAAI Press.