Proceedings of the 22nd International Conference on Principles of Knowledge Representation and Reasoning
KR in Planning and Scheduling

Depth-Bounded Epistemic Planning

Thomas Bolander[1], **Alessandro Burigana**[2], **Marco Montali**[2]
[1]Technical University of Denmark, Denmark
[2]Free University of Bozen-Bolzano, Italy
tobo@dtu.dk, alessandro.burigana@unibz.it, montali@inf.unibz.it

Abstract

We propose a novel algorithm for epistemic planning based on dynamic epistemic logic (DEL). The novelty is that we limit the depth of reasoning of the planning agent to an upper bound b, meaning that the planning agent can only reason about higher-order knowledge to at most (modal) depth b. We then compute a plan requiring the lowest reasoning depth by iteratively incrementing the value of b. The algorithm relies at its core on a new type of "canonical" b-bisimulation contraction that guarantees unique minimal models by construction. This yields smaller states wrt. standard bisimulation contractions, and enables to efficiently check for visited states. We show soundness and completeness of our planning algorithm, under suitable bounds on reasoning depth, and that, for a bound b, it runs in $(b{+}1)$-EXPTIME. We implement the algorithm in a novel epistemic planner, DAEDALUS, and compare it to the EFP 2.0 planner on several benchmarks from the literature, showing effective performance improvements.

1 Introduction

Automated planning is central in AI research (Ghallab, Nau, and Traverso 2004; Geffner and Bonet 2013). In fully observable and deterministic domains, the output of a planning algorithm (a *planner*) is a sequence of actions achieving the desired goal state from a given initial state. *Epistemic planning* is the enrichment of planning with epistemic notions, in particular knowledge and belief, including *higher-order knowledge*, *i.e.*, knowledge about what other agents know, knowledge about what someone knows about someone else, etc. One of the reference frameworks for epistemic planning is dynamic epistemic logic (DEL) (Baltag, Moss, and Solecki 1998) in which epistemic states are represented as Kripke models. *DEL planning* was first pursued by Bolander and Andersen (2011), but independently conceived by others (Löwe, Pacuit, and Witzel 2011; Pardo and Sadrzadeh 2012; Aucher 2012). Recently, a special issue of the journal Artificial Intelligence (AIJ) was devoted to epistemic planning (Belle et al. 2022).

One of the main challenges in epistemic planning is its high computational complexity: unrestricted DEL planning has an undecidable plan existence problem (Bolander and Andersen 2011). This has led researchers to seek more tractable fragments, both theoretically and practically. Intuitively, a main source of complexity is the fact that DEL planning allows agents to reason about others' (higher-order) knowledge up to any nesting level. Thus, limiting reasoning depth seems a promising approach to tame DEL planning's complexity. This led to decidable fragments by, e.g., restricting actions to be propositional, so that the required reasoning depth is bound by the modal depth of the goal formula (Yu, Wen, and Liu 2013), or by imposing axioms on the semantics leading to collapse higher-order reasoning to lower-order (Burigana et al. 2023). An alternative to DEL planning is the *sentential approach*, where states are modelled as knowledge (or belief) bases, *i.e.*, sets of formulas. Here, decidable fragments have been single out by explicitly bounding the modal depth of formulas in states (Muise et al. 2015; Muise et al. 2022).

Our paper is also concerned with taming the computational complexity of epistemic planning via restricting the reasoning bound. However, it is not about achieving a well-behaved, e.g. decidable, fragment of epistemic planning by considering a framework with fixed reasoning bounds. Rather, we propose an alternative algorithmic approach that maintains the generality of unrestricted DEL planning, but always computes a plan using the lowest reasoning depth. This is achieved via an "iterative deepening" algorithm that iteratively increments the allowed reasoning depth until a solution is found. Thus we preserve full generality while still achieving the computational benefits of keeping the reasoning depth as low as possible, hence giving us an algorithm that "degrades gracefully" with the increased reasoning bounds required by epistemically intricate planning tasks. For instance, an agent might activate complex higher-order reasoning when playing an epistemic card game like Hanabi, but not when making coffee. If making coffee does not require any higher-order reasoning at all (*i.e.*, it reasoning about the mental states of other agents is not needed), our algorithm will find a plan at reasoning depth 0, collapsing all the epistemic states into propositional ones.

Our contribution is fourfold: 1. We introduce canonical b-bisimulation contractions as a compact, depth-bounded representation of states; 2. We define an iterative-deepening algorithm for computing plans with the lowest possible modal depth; 3. We show soundness, completeness and complexity results; and 4. We implement the algorithm in our novel epistemic planner, DAEDALUS, and show effective improvements wrt. the EFP 2.0 planner (Fabiano et al. 2020).

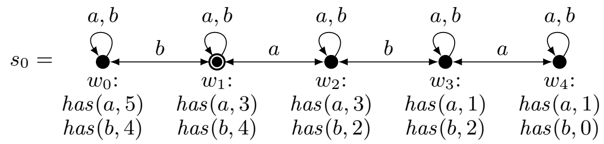

Figure 1: Epistemic state s_0 of Example 1. Bullets represent worlds, labelled by their name and the atoms they satisfy. The actual world is circled. Edges represent the accessibility relations.

2 Preliminaries

Dynamic Epistemic Logic We recall the main concepts of DEL. For a more complete account, see van Ditmarsch, van der Hoek, and Kooi (2007). Let $\mathcal{P}$ be a finite set of *atomic propositions* (*atoms*) and $\mathcal{AG}$ a finite set of *agents*. The language $\mathcal{L}$ of *multi-agent epistemic logic* is given by: $\phi ::= p \mid \neg\phi \mid \phi \wedge \phi \mid \Box_i \phi$ ($p \in \mathcal{P}$, $i \in \mathcal{AG}$). Formula $\Box_i\phi$ is read as "agent i knows/believes that ϕ". Symbols $\vee$, $\rightarrow$ and $\Diamond_i$ are defined by abbreviation as usual.

Definition 1 (States). *An* (epistemic) model *of $\mathcal{L}$ is a triple $M = (W, R, L)$ where: 1) $W \neq \varnothing$ is a finite set of* (possible) worlds*; 2) $R : \mathcal{AG} \rightarrow 2^{W \times W}$ maps to each agent i an* accessibility relation R_i*; 3) $L : W \rightarrow 2^{\mathcal{P}}$ maps to each world w a* label $L(w)$. *An* (epistemic) state *of $\mathcal{L}$ is a pair $s = (M, w)$, where $w \in W$ is the* actual world.

We often write wR_iv for $(w, v) \in R_i$. Truth of formulas $\phi \in \mathcal{L}$ in state (M, w) is defined inductively as follows, with the standard clauses for propositional connectives:

$(M, w) \models p$ iff $p \in L(w)$
$(M, w) \models \Box_i\phi$ iff for all v, if wR_iv then $(M, v) \models \phi$

Example 1 (Consecutive Number Puzzle (van Ditmarsch and Kooi 2015)). *Agents a and b are given two* consecutive *numbers $n_a, n_b \in [0, N]$, only seeing their own number. Let $has(i, n)$ denote that agent i has number n. State s_0 (Figure 1) has $N = 5$, $n_a = 3$ and $n_b = 4$. We have $s_0 \models \Box_b\big(has(b, 4) \wedge (has(a, 3) \vee has(a, 5))\big) \wedge \neg\Box_b has(a, 3) \wedge \neg\Box_b has(a, 5)$: b knows herself to have 4 and knows a to have either 3 or 5, but b doesn't know which of the two.*

In DEL, actions are represented by *event models*. These are Kripke structures on a set of *events*, where each event represents a possible perspective on an action defined via a *precondition* and *postconditions*.

Definition 2 (Actions). *An* event model *of $\mathcal{L}$ is a tuple $A = (E, Q, pre, post)$ where: 1) $E \neq \varnothing$ is a finite set of* events*; 2) $Q : \mathcal{AG} \rightarrow 2^{E \times E}$ maps to each agent i an accessibility relation Q_i; 3) $pre : E \rightarrow \mathcal{L}$ maps to each event a* precondition*; 4) $post : E \times \mathcal{P} \rightarrow \mathcal{L}$ maps to each event-atom pair a* postcondition*. An* action *of $\mathcal{L}$ is a pair $\alpha = (A, e)$ where $e \in E$ is the* actual event.

We often write eQ_if for $(e, f) \in Q_i$. We now introduce the notion of modal depth of an action (sequence), which will play a central role in our algorithm.

Definition 3 (Modal depth). *The* modal depth *of formulas of $\mathcal{L}$ is defined by: $md(p) = 0$ (for $p \in \mathcal{P}$), $md(\neg\phi) = md(\phi)$, $md(\phi_1 \wedge \phi_2) = \max\{md(\phi_1), md(\phi_2)\}$ and $md(\Box_i\phi) = 1 + md(\phi)$. The* modal depth *of an action α is the maximal modal depth of its pre- and postconditions,* i.e., $md(\alpha) = \max\{md(pre(e)), md(post(e, p)) \mid e \in E, p \in \mathcal{P}\}$. *The* modal depth *of $\pi = \alpha_1, \ldots, \alpha_l$ is $md(\pi) = \sum_{i \leq l} md(\alpha_i)$.*

An action is executed in a state via the *product update*.

Definition 4 (Product Update). *Let $s = ((W, R, L), w)$ be a state and $\alpha = ((E, Q, pre, post), e)$ an action. We say that α is* applicable *in s if $s \models pre(e)$, and if so, the* product update *of s with α is the state $s \otimes \alpha = ((W', R', L'), (w, e))$:*

$W' = \{(w, e) \in W \times E \mid (M, w) \models pre(e)\}$
$R'_i = \{((w, e), (v, f)) \in W' \times W' \mid wR_iv \text{ and } eQ_if\}$
$L'((w, e)) = \{p \in \mathcal{P} \mid (M, w) \models post(e, p)\}$.

Example 2. *We can define the* public announcement *of a formula ϕ as the action $ann(\phi) = ((\{e\}, Q, pre, post), e)$ where $Q_i = \{(e, e)\}$ for all i, $pre(e) = \phi$ and $post(e, p) = p$ for all $p \in \mathcal{P}$ (Baltag, Moss, and Solecki 1998). Continuing Example 1, $\alpha = ann(\bigwedge_{0 \leq k \leq N} \neg\Box_b has(a, k))$ is the public announcement of "b doesn't know a's number". The state $s_0 \otimes \alpha$ is achieved by deleting w_4 from s_0: That world represents a situation where b has 0 and hence knows a to have 1, contradicting the announcement.*

We now extend the epistemic language $\mathcal{L}$ with *dynamic modalities* $[\alpha]\phi$, where $\alpha = (A, e)$ is an action. We call the extended language $\mathcal{L}_{dyn}$, and we read $[\alpha]\phi$ as "every execution of α yields a state satisfying ϕ". The semantics is:

$s \models [\alpha]\phi$ iff $s \models pre(e)$ implies $s \otimes \alpha \models \phi$

We often write $[\alpha_1, \ldots, \alpha_l]\phi$ for $[\alpha_1] \ldots [\alpha_l]\phi$ and define the *modal depth* of $[\alpha]\phi$ as $md([\alpha]\phi) = md(\alpha) + md(\phi)$.

Epistemic Planning We recall the notions of epistemic planning tasks and solutions (Aucher and Bolander 2013). For a sequence $\pi = \alpha_1, \ldots, \alpha_l$ of actions and $1 \leq k \leq l$, $\pi_{\leq k}$ denotes the prefix $\alpha_1, \ldots, \alpha_k$ of π, and $s \otimes \pi$ the state $s \otimes \alpha_1 \cdots \otimes \alpha_l$ (if π is empty, this is just s). We say that π is *applicable* in s if for all k, α_k is applicable in $s \otimes \pi_{\leq k-1}$.

Definition 5. *An* (epistemic) planning task *is a triple $T = (s_0, \mathcal{A}, \phi_g)$, where s_0 is a state (the* initial state*), $\mathcal{A}$ is a finite set of actions, and $\phi_g \in \mathcal{L}$ is the* goal formula*. A* solution *(or* plan*) to T is a finite sequence $\pi = \alpha_1, \ldots, \alpha_l$ of actions of $\mathcal{A}$ such that π is applicable in s_0 and $s_0 \models [\pi]\phi_g$.*

Bisimulations We recall bisimulations and bounded bisimulations (Blackburn, Rijke, and Venema 2001).

Definition 6. *Let $b \geq 0$. A* b-bisimulation *between states $s = ((W, R, L), w)$ and $s' = ((W', R', L'), w')$ is a sequence $Z_b \subseteq \cdots \subseteq Z_0 \subseteq W \times W'$ s.t. $(w, w') \in Z_b$ and for all $h < b$:*

[atom] If $(w, w') \in Z_0$, then $L(w) = L(w')$.
[forth$_h$] If $(w, w') \in Z_{h+1}$ and wR_iv, then there exists $v' \in W'$ such that $w'R'_iv'$ and $(v, v') \in Z_h$.
[back$_h$] If $(w, w') \in Z_{h+1}$ and $w'R'_iv'$, then there exists $v \in W$ such that wR_iv and $(v, v') \in Z_h$.

If a b-bisimulation between s and s' exists, s and s' are b-bisimilar*, denoted $s \leftrightarrows_b s'$. When $(M, w) \leftrightarrows_b (M', w')$, we often simply write $w \leftrightarrows_b w'$, and call w and w'* b-bisimilar *(when M, M' are clear from the context). The* b-bisimulation class *of a world w is $[w]_b = \{v \in W \mid v \leftrightarrows_b$*

$w\}$. We say that s and s' are bisimilar, *denoted $s \underline{\leftrightarrow} s'$, if there exists a single relation Z satisfying the conditions above with the subscript on Z removed everywhere.*

Note that $s \underline{\leftrightarrow}_b s'$ implies $s \underline{\leftrightarrow}_h s'$ for all $h < b$ (similarly for $w \underline{\leftrightarrow}_b w'$). Given a set $\Phi \subseteq \mathcal{L}$ and states s, s', we say that s and s' *agree on* Φ if, for all $\phi \in \Phi$, $s \models \phi$ iff $s' \models \phi$.

Proposition 1 (Blackburn, Rijke, and Venema 2001)**.** *Two finite states are bisimilar iff they agree on $\mathcal{L}$. They are b-bisimilar iff they agree on $\{\phi \in \mathcal{L} \mid md(\phi) \leq b\}$.*

Proposition 2 (van Ditmarsch, van der Hoek, and Kooi 2007)**.** *Let $s \underline{\leftrightarrow} s'$ and let α be applicable in both s and s'. Then, $s \otimes \alpha \underline{\leftrightarrow} s' \otimes \alpha$.*

The following relatively recent result will play a central role in our paper. Together with Proposition 1, it gives a lower bound on the modal depth of formulas whose truth is preserved after product updates (after action executions).

Proposition 3 (Bolander and Lequen 2022)**.** *Let $s \underline{\leftrightarrow}_b s'$ and let π be an action sequence applicable in both s and s' with $md(\pi) \leq b$. Then, $s \otimes \pi \underline{\leftrightarrow}_{b-md(\pi)} s' \otimes \pi$.*

The cited source only formulates the result for individual actions (action sequences of length 1), but it generalizes immediately. We now clarify the relevance of these results. The modal depth of formulas formalizes the informal notion of reasoning depth used in the introduction: e.g., reasoning on the statement "a knows that b knows neither agent has 0" requires a formula modal depth 2. Proposition 1 shows that b-bisimilar states agree on all formulas up to modal depth b, so an agent limited to reasoning depth b cannot distinguish between them. Thus, the agent's internal state could be anyone of a given b-bisimulation class, e.g., a minimal one (see next section for an elaboration on this). However, reasoning depth is not necessarily preserved under product updates. For example, if s'_0 is obtained from s_0 (Figure 1) by deleting w_3 and w_4, then $s_0 \underline{\leftrightarrow}_1 s'_0$, since formulas of modal depth ≤ 1 are evaluated only in the actual world w_1 and its immediate neighbours, but $s_0 \otimes \alpha \underline{\leftrightarrow}_1 s'_0 \otimes \alpha$ does *not* hold when α is the public announcement that "b doesn't know a's number" of Example 2. This action will remove the rightmost world no matter whether it is applied to s_0 or s'_0, so when applied to s'_0, it will remove w_2, and then e.g. $\Diamond_a has(a, 3)$ will be true in $s_0 \otimes \alpha$ but not in $s'_0 \otimes \alpha$. So if an agent needs to be able to reason to depth 1 about the state resulting from executing α, it needs to be able to reason to at least to depth 2 about the state before the action. Proposition 3 then guarantees that reasoning to depth 2 about the former state suffices, since $md(\alpha) = 1$. This is key in depth-bounded epistemic planning, since planning is reasoning about possible future states, and depth-bounded reasoning to depth b corresponds to b-bisimulation invariance.

Proposition 4. *If two finite states s and s' are b-bisimilar, they agree on $\{\phi \in \mathcal{L}_{dyn} \mid md(\phi) \leq b\}$.*[1]

Proof sketch. The proof is by induction on the structure of ϕ, where the cases of the connectives and modalities in $\mathcal{L}$ are exactly as in the proof of Proposition 1, and the case of the dynamic modalities $[\alpha]\psi$ follows from Proposition 3 and the definition of $md([\alpha]\psi)$. □

[1]Full proofs are available in the Appendix of the extended version available at https://arxiv.org/abs/2406.01139.

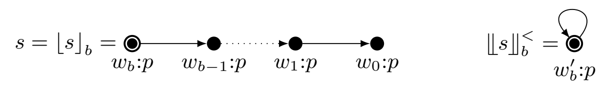

Figure 2: Standard ($\lfloor s \rfloor_b$) and rooted ($\llbracket s \rrbracket_b^<$) b-contractions of s (Definition 9). As we have only one agent, we omit agent labels.

3 Rooted and Canonical b-Contractions

A significant challenge for epistemic planners is to handle the rapid growth of the size of states following updates (Bolander and Andersen 2011), even when minimizing states using bisimulation contractions (Yu, Wen, and Liu 2013; Fabiano et al. 2020; Bolander, Dissing, and Herrmann 2021a). Here we seek to model agents with a bound on their reasoning depth (a bound on the modal depth of the formulas they can reason with). We achieve this by using b-bisimulation contractions, abbreviated *b-contractions*. These were introduced to epistemic planning by Yu, Wen, and Liu (2013), defining the b-contraction $\lfloor s \rfloor_b$ of a state $s = ((W, R, L), w)$ as the quotient structure of s wrt. $\underline{\leftrightarrow}_b$, *i.e.*, $\lfloor s \rfloor_b = ((W', R', L'), [w]_b)$ with $W' = \{[w]_b \mid w \in W\}$, $R'_i = \{([w]_b, [v]_b) \mid wR_iv\}$, and $L'([w]_b) = L(w)$. We call this the *standard b-contraction* of s. Unfortunately, $\lfloor s \rfloor_b$ is not always minimal (*i.e.*, it does not have the smallest number of worlds and edges) among the states b-bisimilar to s, as the next example shows (Bolander and Burigana 2024).

Example 3. *Consider the chain state s in Figure 2, left. Since p is true in all worlds, and the length of the chain is b, a minimal state b-bisimilar to s is a singleton state with a loop (Figure 2, right). This is because the loop state preserves all formulas up to depth b, cf. Proposition 1. However, the standard b-contraction of s is simply s itself, as no two worlds of s satisfy the same formulas up to depth b.*

In the following, $b \geq 0$ denotes a constant (a *bound*), $s = (M, w_d)$ a state with $M = (W, R, L)$, $<$ a total order on W.

Rooted b-contractions. Recently, Bolander and Burigana (2024) developed a novel type of b-contractions, called *rooted b-contractions*, that guarantee minimality of contracted states. We briefly state the definition and minimality result from that paper (Definitions 7–9 and Theorem 1 below), and refer the reader to the paper for further details.

Definition 7 (Depth and Bound)**.** *The* depth $d(w)$ *of a world $w \in W$ is the length of the shortest path from w_d to w (∞ if no such path exists). The* bound *of w is $b(w) = b - d(w)$.*

Definition 8. *Let $x, y \in W$ with $b(x), b(y) \geq 0$. We say that x* represents *y, denoted $x \succeq y$, if $b(x) \geq b(y)$ and $x \underline{\leftrightarrow}_{b(y)} y$. If furthermore $b(x) > b(y)$, we say that x* strictly represents *y, denoted by $x \succ y$. The set of* maximal representatives *of W is the set $W^{\max} = \{x \in W \mid b(x) \geq 0$ and $\neg\exists y(y \succ x)\}$. We also let $W^{\max}_{>0} = \{x \in W^{\max} \mid b(x) > 0\}$. For $h \leq b$, the* least h-representative *of w is the world $\min^<_h(w) = \min_<\{v \in W^{\max} \mid v \underline{\leftrightarrow}_h w\}$. The* representative class *of $w \in W$ is $[w]_{b(w)}$, compactly denoted $\llbracket w \rrbracket$.*

Definition 9. *The* rooted b-contraction *of s is* $\llbracket s \rrbracket^{<}_{b} = ((W', R', L'), \llbracket w_d \rrbracket)$*, where:*
$W' = \{\llbracket x \rrbracket \mid x \in W^{\max}\}$
$R'_i = \{(\llbracket x \rrbracket, \llbracket \min^{<}_{b(x)-1}(y) \rrbracket) \mid x \in W^{\max}_{>0}$ *and* $xR_iy\}$
$L'(\llbracket x \rrbracket) = L(x)$*, for all* $\llbracket x \rrbracket \in W'$.

Theorem 1 (Bolander and Burigana 2024). $\llbracket s \rrbracket^{<}_{b}$ *is a minimal state b-bisimilar to s,* i.e.*, it has a minimal number of worlds and edges among all states b-bisimilar to s.*

Canonical b-contractions. Below, we devise a novel planning algorithm using *graph search* (Russell and Norvig 2010) to find plans. As we want to limit the reasoning depth of the planning agent to some bound b, we can replace each state s in the search tree by $\llbracket s \rrbracket^{<}_{b}$, since Proposition 1 and Theorem 1 together give us that $\llbracket s \rrbracket^{<}_{b}$ is a minimal state preserving the truth of formulas up to depth b in s. Due to the minimality of rooted b-contractions, this would guarantee minimality of the representation of each state in the search tree. However, when using graph search for planning, we also need to be able to efficiently check whether a computed state is already in the search tree. In our case, this would amount to checking b-bisimilarity between the computed state and existing states in the search tree, which is costly. Bolander, Dissing, and Herrmann (2021a) found a solution to this problem in the case of standard bisimulations by devising a so-called *ordered partition refinement* algorithm for computing bisimulation contractions guaranteeing two bisimilar states to have *identical* contractions, hence reducing bisimilarity checks to identity checks. Inspired by this work, we define a notion of *canonical b-contractions* guaranteeing two b-bisimilar states to have identical contractions, in turn providing fast state comparisons for our planning algorithm. Before giving the definition, we illustrate the issue of rooted b-contractions with an example.

Example 4. *Let s be the state in Figure 3, let t be a state only differing from s by swapping the names of w_1 and w_2 (Figure 4), let the total order $<$ on W (the world set of s and t) be given by $w_i < w_j$ iff $i < j$, and let $b = 3$. Then $b(w_0) = 3$, $b(w_1) = b(w_2) = 2$, and $b(w_3) = b(w_4) = 1$. Since in both states no two worlds are 1-bisimilar, all worlds of W are maximal representatives, so $\llbracket s \rrbracket^{<}_{3}$ and $\llbracket t \rrbracket^{<}_{3}$ (Figures 5 and 6) will contain a world $\llbracket w \rrbracket = [w]_{b(w)}$ for each $w \in W$. Note that both 3-contracted states only differ from their original model by the deletion of a different "unnecessary" edge (as this depends on $<$). Hence, despite s and t being isomorphic, $\llbracket s \rrbracket^{<}_{3}$ and $\llbracket t \rrbracket^{<}_{3}$ are not! This shows that rooted b-contractions do not ensure unique representatives of classes of b-bisimilar states.*

Definition 10. *The* h-signature $\sigma_h(w)$ *of a world w is the pair $(L(w), \Sigma_h(w))$, where the function $\Sigma_h(w)$ maps to each agent i a set $\Sigma_h(w, i) = \{\sigma_{h-1}(v) \mid wR_iv\}$ of $(h-1)$-signatures, if $h > 0$, and $\varnothing$ otherwise. We call $\sigma_{b(w)}(w)$ the* representative signature *of w, compactly denoted $\sigma(w)$.*

In the h-signature of w, the label $L(w)$ describes what atoms hold at w and an $(h-1)$-signature $\sigma_{h-1}(v) \in \Sigma_h(w, i)$ represents agent i's knowledge/beliefs at v up to depth $h-1$. All $(h-1)$-signatures map to each agent a set of $(h-2)$-signatures, and so on until 0-signatures, which only contain worlds labels. Hence, the h-signature of w captures the model structure up to depth h from w. This will allow us to prove that two worlds are h-bisimilar iff they have the same h-signature (Lemma 1). In what follows, $h \geq 0$ denotes a constant, and $M = (W, R, L)$ and $M' = (W', R', L')$ two models with $x \in W$ and $x' \in W'$.

Lemma 1. $x \leftrightarroweq_h x'$ *iff* $\sigma_h(x) = \sigma_h(x')$.

Proof sketch. ($\Rightarrow$) Induction on h. For $h = 0$, $x \leftrightarroweq_0 x'$ iff $L(x) = L'(x')$ iff $\sigma_0(x) = \sigma_0(x')$. For $h > 0$, $x \leftrightarroweq_h x'$ implies $L(x) = L'(x')$, and for all $i \in \mathcal{AG}$, the sets of $(h-1)$-signatures of i-accessible worlds are equal (i.h. and Definition 10). ($\Leftarrow$) Assume $\sigma_h(x) = \sigma_h(x')$. Letting $Z_g = \{(w, w') \mid \sigma_g(w) = \sigma_g(w')\}$ for $g \leq h$, $Z_h, \ldots, Z_0$ is an h-bisimulation between x and x' (i.h. and Definition 10). $\square$

As $\mathcal{P}$ and $\mathcal{AG}$ are fixed finite sets, we can assume a fixed total order on them. This induces a fixed total order $\lessdot$ on signatures as in Bolander, Dissing, and Herrmann (2021a). From this, we define the notion of *canonical signatures*.

Definition 11. *The* canonical signature to depth h *($h \leq b$) of a world $w \in W$ is the representative signature $\sigma^{\star}_{h}(w) = \min_{\lessdot}\{\sigma(v) \mid v \in W^{\max}$ and $\sigma_h(v) = \sigma_h(w)\}$.*

Definition 12. *The* canonical b-contraction *of s is the state* $\llbracket s \rrbracket^{\star}_{b} = ((W', R', L'), \sigma(w_d))$*, where:*
$W' = \{\sigma(x) \mid x \in W^{\max}\}$
$R'_i = \{(\sigma(x), \sigma^{\star}_{b(x)-1}(y)) \mid x \in W^{\max}_{>0}$ *and* $xR_iy\}$
$L'(\sigma(x)) = L(x)$*, for all* $\sigma(x) \in W'$.

Note that canonical b-contractions only differ from rooted b-contractions by the naming of worlds ($\sigma(x)$ instead of $\llbracket x \rrbracket$) and by the choice of representative of the class of worlds that are $(b(x)-1)$-bisimilar to y in the definition of R'_i (using $\sigma^{\star}_{b(x)-1}(y)$ instead of $\llbracket \min^{<}_{b(x)-1}(y) \rrbracket$). This means that the proof of Theorem 1 carries directly over to this setting:

Theorem 2. $\llbracket s \rrbracket^{\star}_{b}$ *is a minimal state b-bisimilar to s.*

Furthermore, canonical b-contractions enjoy the following identity property:

Theorem 3. $s \leftrightarroweq_b t$ *iff* $\llbracket s \rrbracket^{\star}_{b} = \llbracket t \rrbracket^{\star}_{b}$.

Proof sketch. ($\Leftarrow$): by Theorem 2. ($\Rightarrow$): We use Lemma 1 and Definitions 11–12 to show that $\llbracket s \rrbracket^{\star}_{b}$ and $\llbracket t \rrbracket^{\star}_{b}$ have the same worlds and accessibility relations, and Definition 12 to show that they have the same labels. $\square$

Example 5. *Theorem 3 ensures that b-bisimilar states have identical canonical b-contractions, unlike rooted contractions where this might not even hold for isomorphic states (cf. Example 4). We identified the problem to be that the choice of "unnecessary" edges to be deleted depended on the total order on W,* i.e.*, on the naming of worlds. Canonical b-contractions solve this problem, as the choice of "representative edges" is uniquely determined by the signatures, which are naming independent. In particular, referring again to Example 4, since $b(w_3) = 1$ in s when $b = 3$, the edge (w_3, w_2) of s is replaced by $(\sigma(w_3), \sigma^{\star}_{0}(w_2))$ in $\llbracket s \rrbracket^{\star}_{b}$, where $\sigma^{\star}_{0}(w_2) = \min_{\lessdot}\{\sigma(v) \mid v \in W^{\max}, \sigma_0(v) =$*

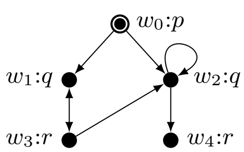

Figure 3: State s.

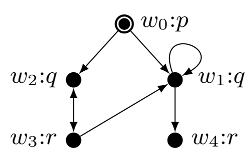

Figure 4: State t.

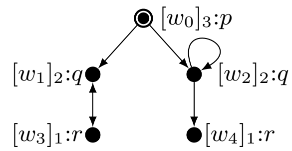

Figure 5: State $\llbracket s \rrbracket_3^{<}$.

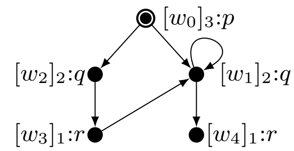

Figure 6: State $\llbracket t \rrbracket_3^{<}$.

$\sigma_0(w_2)\} = \min_{\lessdot}\{\sigma(v) \mid v \in W^{\max}, L(v) = L(w_2) = \{q\}\}$*, showing that the end node of the edge only depends on the fixed total order on signatures and the label of worlds.*

4 Iterative Bound-Deepening Search

Unlike more typical search strategies aimed at computing a plan that is as short as possible, our goal is to find a plan with a modal depth as low as possible (Definition 3). To this end, we impose a bound $\mathbf{b}$ on the reasoning depth of the planning agent (the modal depth of formulas it can reason about), which in turn allows us to use $\mathbf{b}$-contractions to reduce state sizes. To make the intution underlying our algorithm clearer, we first introduce some new helpful notions and corresponding results.

Definition 13. *A state s' is called an* exact representation *of a state s if $s' \leftrightarroweq s$. The state s' is called an* approximate representation *(or simply* approximation*) of s if $s' \leftrightarroweq_b s$ for some $b \geq 0$. In this case, s' is also called a b-*approximation *of s. If s' is an approximate, but not an exact, representation of s, we call it a* proper approximation*.*

Proposition 5. *If s is an exact representation of s', then the two states agree on $\mathcal{L}_{dyn}$. If s is a b-approximation of s', then they agree on all $\mathcal{L}_{dyn}$-formulas of modal depth at most b. In particular, given a state s, its canonical b-contraction $\llbracket s \rrbracket_b^{\star}$ is a b-approximation of s and agrees with it on all $\mathcal{L}_{dyn}$-formulas of modal depth at most b.*

Proof. The first statement follows directly from Proposition 1, as any formula of $\mathcal{L}_{dyn}$ is equivalent to a formula in $\mathcal{L}$ (van Ditmarsch, van der Hoek, and Kooi 2007). The second statement follows from Proposition 4. The third statement follows from the second statement and Theorem 2. $\square$

To check whether an action sequence π is a solution to a planning task $(s_0, \mathcal{A}, \phi_g)$, we need to check whether $s_0 \models [\pi]\phi_g$ (Definition 5). Proposition 5 gives us that this is equivalent to checking whether $\llbracket s_0 \rrbracket^{\star}_{md([\pi]\phi_g)} \models [\pi]\phi_g$. Thus to check whether a particular action sequence π is a solution, we can always replace the true initial state s_0 by $\llbracket s_0 \rrbracket^{\star}_{md([\pi]\phi_g)}$, or by any other $\llbracket s_0 \rrbracket^{\star}_{\mathbf{b}}$ with $\mathbf{b} \geq md([\pi]\phi_g)$. This will be exploited in our algorithm below. The algorithm starts out with an initial value of the bound $\mathbf{b}$ and attempts to compute a solution from the $\mathbf{b}$-contracted initial state $\llbracket s_0 \rrbracket^{\star}_{\mathbf{b}}$. If this fails, $\mathbf{b}$ is incremented, and a new search is performed.

4.1 Description of the Planning Algorithm

We now describe our planning Algorithm 1 called Iterative Bound-Deepening Search (IBDS). The algorithm builds a search tree. Each node of the tree contains a contracted state $\llbracket s \rrbracket^{\star}_b$ representing some true state s. Thus each node state is an approximate representation of the corresponding true state (Definition 13 and Proposition 5). Formally, a *node* of the search tree is a pair $n = (s, b)$, where s is the state of n (denoted n.state) and b is the *(depth) bound* (denoted n.bound). The bound b is intended to guarantee that n.state is at least a b-approximation of the corresponding true state, and hence agrees with all formulas of the true state up to modal depth b (Proposition 5). Later, in Lemma 2, we formally prove that this property holds for all nodes in the search tree. For now, we only provide informal arguments.

To compute $\llbracket s \rrbracket^{\star}_b$, we use *bounded partition refinements* (Bolander and Lequen 2022), a variation of standard partition refinements (Paige and Tarjan 1987). The algorithm manipulates via *refinement* operations a partition P_0 of W (initially calculated wrt. labels): for each element $B \in P_0$, called a *block*, and for each relation R_i, a refinement produces a new partition P_1 by splitting the worlds of B wrt. which blocks of P_0 they can access via R_i. Refinements are applied recursively until the sequence $[P_0, \ldots, P_b]$ of partitions is computed. This gives the bounded bisimulation classes on W: the blocks of P_h are the h-bisimulation classes of W (Bolander and Lequen 2022, Prop. 7). Canonical b-contractions are then obtained from the partitions by following Definition 12.

In IBDS (line 1), the integer $\mathbf{b} \geq 0$ denotes the global maximum modal depth of the formulas that we evaluate. Initially, we let $\mathbf{b} = md(\phi_g)$ (line 2). We then iteratively call the BOUNDEDSEARCH algorithm over increasing values of $\mathbf{b}$ until a plan π is found (lines 2-4). BOUNDEDSEARCH uses breadth-first search (BFS) starting from the initial node $n_0 = \text{INITNODE}(s_0, \mathbf{b}) = (\llbracket s_0 \rrbracket^{\star}_{\mathbf{b}}, \mathbf{b})$ (line 6). In the first call to BOUNDEDSEARCH, $\mathbf{b}$ has value $md(\phi_g)$, implying that the state $\llbracket s_0 \rrbracket^{\star}_{\mathbf{b}}$ of the root note n_0 at least $md(\phi_g)$-approximates the true initial state s_0. This is sufficient to check whether the goal formula is satisfied in the true initial state s_0, since the root state $\llbracket s_0 \rrbracket^{\star}_{\mathbf{b}}$ and s_0 will agree on ϕ_g (Proposition 5). But we are not guaranteed that they will also agree on $[\pi]\phi_g$, for non-empty action sequences π.

We keep track of the nodes to be expanded in the *frontier* queue, and of the states encountered during search in the *visited* set. While *frontier* contains a node $n = (s, b)$, we extract it from the queue, mark its state as visited and check whether it satisfies the goal formula. If it does, we return the plan that led to s (line 11). Otherwise, we process n by generating child nodes n' for all actions α with $b \geq md(\alpha) + md(\phi_g)$ that are applicable in s (lines 12-14). Each such child node is initialized as $n' = \text{CHILDNODE}((s, b), \alpha) = \text{INITNODE}(s \otimes \alpha, b - md(\alpha)) = (\llbracket s \otimes \alpha \rrbracket^{\star}_{b-md(\alpha)}, b - md(\alpha))$ (lines 17-20). If n'.state was

Algorithm 1 Iterative Bound-Deepening Search

1: **function** IBDS($(s_0, \mathcal{A}, \phi_g)$)
2: **for** $\mathbf{b} \leftarrow md(\phi_g)$ **to** ∞ **do**
3: $\pi \leftarrow$ BOUNDEDSEARCH($(s_0, \mathcal{A}, \phi_g), \mathbf{b}$)
4: **if** $\pi \neq$ *fail* **then return** π
5: **function** BOUNDEDSEARCH($(s_0, \mathcal{A}, \phi_g), \mathbf{b}$)
6: $frontier \leftarrow \langle$INITNODE($s_0$, $\mathbf{b}$)$\rangle$
7: $visited \leftarrow \varnothing$
8: **while** $\neg frontier$.EMPTY() **do**
9: $(s, b) \leftarrow frontier$.POP()
10: $visited$.PUSH(s)
11: **if** $s \models \phi_g$ **then return** plan to s
12: **for all** $\alpha \in \mathcal{A}$ such that $b \geq md(\alpha) + md(\phi_g)$ **do**
13: **if** α is applicable in s **then**
14: $n' \leftarrow$ CHILDNODE((s, b), α)
15: **if** n'.state $\notin visited$ **then** $frontier$.PUSH(n')
16: **return** *fail*
17: **function** INITNODE(s, b)
18: **return** $\left(\llbracket s \rrbracket_b^\star, b\right)$
19: **function** CHILDNODE((s, b), α)
20: **return** INITNODE($s \otimes \alpha$, $b - md(\alpha)$)

Algorithm 2 Improved initialization and child generation

1: **function** INITNODE(s, b, was_bisim)
2: **return** $\left(\llbracket s \rrbracket_b^\star, b, (\llbracket s \rrbracket_b^\star \leftrightarroweq s) \wedge was\_bisim\right)$
3: **function** CHILDNODE((s, b, is_bisim), α)
4: **if** is_bisim **then return** INITNODE($s \otimes \alpha$, b, $true$)
5: **else return** INITNODE($s \otimes \alpha$, $b - md(\alpha)$, $false$)

not previously visited, we push n' onto *frontier* and continue the search (line 15). Since n'.state is a canonically contracted state, Theorem 3 implies that to verify if n'.state has already been visited, it suffices to check whether n'.state $\in visited$. Consider now any added child node n'. Since we intended for the bound b of a node to guarantee that it at least b-approximates the corresponding true state, we need to show that this is preserved when moving from n to n'. In other words, we need to guarantee that if s is a b-approximation of some true state t, then n'.state is an n'.bound-approximation of $t \otimes \alpha$. If s is a b-approximation of t, then by Proposition 3, $s \otimes \alpha$ is a $(b - md(\alpha))$-approximation of $t \otimes \alpha$. Theorem 2 then gives that also n'.state $= \llbracket s \otimes \alpha \rrbracket^\star_{b-md(\alpha)}$ is a $(b - md(\alpha))$-approximation of $t \otimes \alpha$, and as n'.bound $= b - md(\alpha)$, this shows the required. Note furthermore that as $b \geq md(\alpha) + md(\phi_g)$, then n'.bound $\geq md(\phi_g)$. Since n'.state is n'.bound-approximating the true child state $t \otimes \alpha$, this guarantees that n'.state agrees with the true child state on the goal formula, and hence that we can correctly verify whether the goal has been achieved after executing α by checking its truth-value in n'.state.

Finally, if the frontier is empty and no plan was found,we return *fail* and proceed to the next iteration of IBDS.

As seen, the IBDS strategy attempts to compute plans with the lowest possible modal depth (lowest value of $\mathbf{b}$). Moreover, by keeping track of node bounds we can reduce the size of visited states via bounded contractions. As we now show, this reduction can be further improved by finding lower bounds for contractions.

4.2 Improving Bounds for Contractions

We now show how to achieve tighter bounds for contractions, further improving the algorithm. As explained above, a state s of the search tree is guaranteed to b-approximate the corresponding true state t. Sometimes, however, it might be that s is even an exact representation of t, *i.e.*, bisimilar to it. In other words, bounded contractions do not always yield proper approximations (Definition 13). We use this key observation to improve our algorithm as follows. We now let a node be a *triple* $n = (s, b, is\_bisim)$, where the b-contracted state $s = \llbracket t \rrbracket_b^\star$ (t being the true state) and b are as above, and is_bisim (denoted n.is_bisim) is a boolean representing whether $\llbracket t \rrbracket_b^\star \leftrightarroweq t$ holds, *i.e.*, whether the state of a node is an exact representation of the true state (is_bisim true) or not (is_bisim false). We exploit this new information to improve functions INITNODE and CHILDNODE (Algorithm 2). INITNODE now takes an extra boolean parameter was_bisim being true if all states of the antecedents of the node being initialized are exact representations, and it initializes a new node n' together with its bisimilarity information (line 2). We then initialize the frontier with $\langle$INITNODE(s_0, $\mathbf{b}$, $true$)$\rangle$ in line 6 of BOUNDEDSEARCH. CHILDNODE now generates a child node n', from a node $n = (s, b, is\_bisim)$ and an action α, as follows. If is_bisim is true (line 4), then the node state s is bisimilar to the true state t, and hence also $s \otimes \alpha \leftrightarroweq t \otimes \alpha$, by Proposition 2. Thus also the state of the child node is an exact representation of the corresponding true state, and we can hence preserve the current bound by letting n'.bound $= b$. If is_bisim is false (line 5), then we only know that s is a b-approximation of the true state and we proceed as in Algorithm 1.

We can think of the new algorithm as having two different modes for generating child nodes: An *exact mode* – invoked if in the path leading to the new node all states are exact representations, and an *approximate mode* – when some of the states are approximations. The algorithm starts in exact mode and, while it runs in this mode, the bound is preserved, as we are guaranteed that no information is lost due to bounded contractions. As soon as we generate an approximate state, we enter into the approximate mode and continue the search as in Algorithm 1. To avoid ambiguity, from now on we refer to Algorithm 1 as APPROX-IBDS and to the improved version as MIXED-IBDS.

MIXED-IBDS improves APPROX-IBDS in several ways. First, the number of iterations in MIXED-IBDS is reduced in general. Consider a path $n_0 \xrightarrow{\alpha_1} \cdots \xrightarrow{\alpha_l} n_l$ visited by BOUNDEDSEARCH($T, \mathbf{b}$), and let $n_k = (s_k, b_k, \mu_k)$ for each $0 \leq k \leq l$. In APPROX-IBDS, we have $b_k = \mathbf{b} - \sum_{h \leq k} md(\alpha_h)$, as we subtract the modal depth of each applied action from the initial bound (line 20). In MIXED-IBDS this does not necessarily happen for each action, so MIXED-IBDS will generally be able to expand more nodes and add

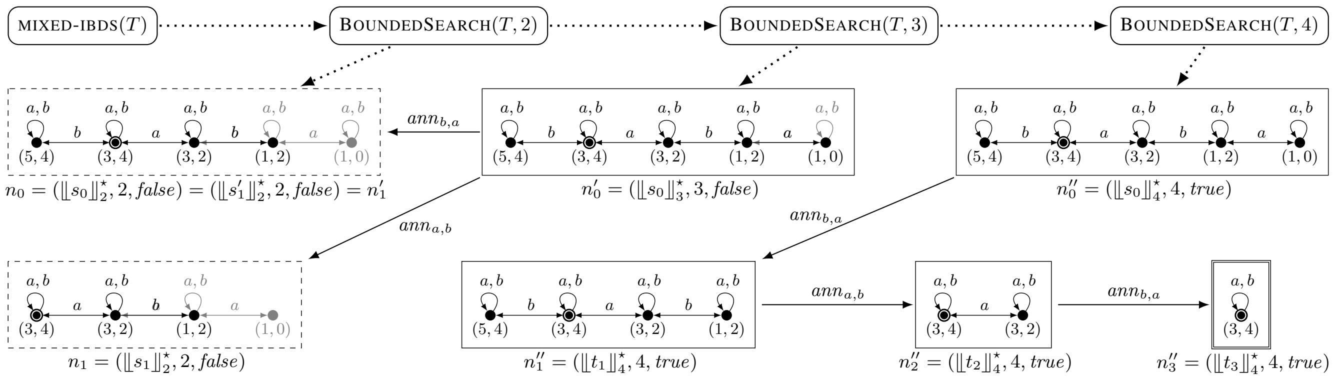

Figure 7: Partial execution of MIXED-IBDS on the planning task T of Example 6. Gray worlds and edges denote the parts of a state that are removed due to bounded contractions (*e.g.*, the rightmost world of s_0 is not included in $\llbracket s_0 \rrbracket_2^\star$). For brevity, each world is labelled by a pair (x, y), meaning that in that world agents a and b have numbers x and y, respectively. Dashed borders denote nodes that can not be expanded with any action, and double borders denote nodes that satisfy the goal.

more children without increasing the bound $\mathbf{b}$, hence generally require fewer iterations of BOUNDEDSEARCH. Second, we improve bounds for contractions: As less iterations are needed, the global bound $\mathbf{b}$ is smaller, so we can take b-contractions with lower values of b, further reducing the size of states. Finally, we implemented an important optimization in MIXED-IBDS (omitted in the pseudocode for readability): We preserve all nodes n of the search tree with n.is_bisim $=$ true across different iterations of MIXED-IBDS, as in each iteration they would simply be recomputed as in the previous ones. As we show in our experiments, this optimization avoids redundant computation and provides effective improvements wrt. APPROX-IBDS.

Example 6. *We now show the execution of* MIXED-IBDS *on the planning task $T = (s_0, \{ann_{a,b}, ann_{b,a}\}, \phi_g)$, where s_0 is from Figure 1, $ann_{i,j} = ann(\bigwedge_{0 \leq k \leq N} \neg \Box_i has(j,k))$ (Example 2) is the public announcement of the fact that agent i does not know j's number, and $\phi_g = \Box_b \Box_a has(b, 4)$. The algorithm invokes* BOUNDEDSEARCH *with increasing reasoning bounds $\mathbf{b}$, starting from $\mathbf{b} = md(\phi_g) = 2$.*

$\mathbf{b} = 2$: *The algorithm performs a BFS starting from the initial node n_0 $=$* INITNODE$(s_0, 2, \mathit{true})$. INITNODE *first computes the canonical 2-contraction of s_0, represented by n_0 in Figure 7. By Proposition 1, $s_0 \not\leftrightarrows \llbracket s_0 \rrbracket_2^\star$ since they disagree on $\Diamond_a \Diamond_b \Diamond_a has(b, 0)$, so we get $n_0 = (\llbracket s_0 \rrbracket_2^\star, 2, \mathit{false})$. After extracting n_0 from the frontier (line 9), since $\llbracket s_0 \rrbracket_2^\star \not\models \phi_g$ and both actions have modal depth 1, the condition $\mathbf{b} \geq md(\alpha) + md(\phi_g)$ (line 12) fails, so no children are generated and the search continues.*

$\mathbf{b} = 3$: *With the same reasoning as above, we get $n'_0 = (\llbracket s_0 \rrbracket_3^\star, 3, \mathit{false})$. Since $\llbracket s_0 \rrbracket_3^\star \not\models \phi_g$ and both actions are applicable in $\llbracket s_0 \rrbracket_3^\star$ satisfying $\mathbf{b} \geq md(\alpha) + md(\phi_g)$, we generate $n_1 = (\llbracket s_1 \rrbracket_2^\star, 2, \mathit{false})$ and $n'_1 = (\llbracket s'_1 \rrbracket_2^\star, 2, \mathit{false})$, where $s_1 = \llbracket s_0 \rrbracket_3^\star \otimes ann_{a,b}$ and $s'_1 = \llbracket s_0 \rrbracket_3^\star \otimes ann_{b,a}$. As shown in Example 2, public announcements delete worlds that do not satisfy the announced formula. Thus, $ann_{a,b}$ removes the leftmost world from $\llbracket s_0 \rrbracket_3^\star$ (as in that world agent a knows that b has number 4), and $ann_{b,a}$ deletes the rightmost one. Note that $\llbracket s'_1 \rrbracket_2^\star = \llbracket s_0 \rrbracket_2^\star$, so n_0 and n'_1 are identical; in Figure 7 we merge them for clarity, despite being separate nodes. Both n_1 and n'_1 have bound 2, which is insufficient to proceed (as previously shown), so the search advances to the next iteration.*

$\mathbf{b} = 4$: *Due to space constraints, we only describe the path of the search tree induced by the action sequence $\pi = ann_{b,a}, ann_{a,b}, ann_{b,a}$ (shown in Figure 7). Since $\llbracket s_0 \rrbracket_4^\star$ (Figure 7) and s_0 are isomorphic, they are also bisimilar, so the initial node of the path is $n''_0 = (\llbracket s_0 \rrbracket_4^\star, 4, \mathit{true})$. Following the computation of the path, we can show that the reasoning bound is sufficiently high to visit all nodes (updates are computed as in the previous iteration). It can also be shown that no other previously visited path leads to a goal state. Thus, node n''_3 is going to be eventually visited, and since n''_3.state $\models \phi_g$ the action sequence π is returned.*

5 Soundness, Completeness and Complexity

In the following, $T = (s_0, \mathcal{A}, \phi_g)$ is a planning task and $\mathbf{b} \geq md(\phi_g)$ a constant. For a sequence $\pi = \alpha_1, \dots, \alpha_l$ of actions, $path(\pi)$ denotes a path $n_0 \xrightarrow{\alpha_1} \cdots \xrightarrow{\alpha_l} n_l$ in the graph visited by BOUNDEDSEARCH$(T, \mathbf{b})$.

Lemma 2. *Let $n_k = (t_k, b_k, \mu_k)$ be the last node of $path(\pi)$, where $\pi = \alpha_1, \dots, \alpha_k$. Then:*
(1) If $\mu_k = \mathit{true}$, then $t_k \leftrightarrows s_0 \otimes \pi$;
(2) If $\mu_k = \mathit{false}$, then $t_k \leftrightarrows_{b_k} s_0 \otimes \pi$;
(3) $b_k \geq \mathbf{b} - \Sigma_{i \leq k} md(\alpha_i)$ and $b_k \geq md(\phi_g)$.

Proof sketch. By induction on k. Base case ($k = 0$): Immediate by construction, Proposition 5, and $\mathbf{b} \geq md(\phi_g)$. Induction ($k > 0$): Assume the properties for n_{k-1}. If μ_{k-1} is true, Proposition 2 gives (1-2), and the bound is unchanged. Otherwise, (1) is trivial, (2) follows by Proposition 3, and the bound decreases by $md(\alpha_k)$, preserving (3). ☐

Theorem 4 (Soundness)**.** *If* BOUNDEDSEARCH$(T, \mathbf{b})$ *returns an action sequence π, then π is a solution to T.*

Proof. Let $n_k = (t_k, b_k, \mu_k)$ be the last node of $path(\pi)$. From line 11 of the algorithm, we get $t_k \models \phi_g$. Since $t_k \leftrightarrows_{b_k} s_0 \otimes \pi$ and $b_k \geq md(\phi_g)$ (by Lemma 2), Proposition 1 gives us that t_k and $s_0 \otimes \pi$ agree on formulas up to depth $md(\phi_g)$, and hence $s_0 \otimes \pi \models \phi_g$. Thus, π is a solution to T. $\square$

We now present the parameters used for our completeness and complexity results, originally introduced in the context of epistemic plan verification (van de Pol, van Rooij, and Szymanik 2018; Bolander and Lequen 2022). We let $\mathsf{a} = |\mathcal{AG}|$, $\mathsf{p} = |\mathcal{P}|$, $\mathsf{c} = \max\{md(\alpha) \mid \alpha \in \mathcal{A}\}$, $\mathsf{o} = md(\phi_g)$, and u be the maximal allowed solution length. We use $|T|$ for the size of T, *i.e.*, the sum of the sizes of s_0, $\mathcal{A}$ and ϕ_g.

Theorem 5 (Completeness)**.** *If T has a solution of length u, then* BOUNDEDSEARCH$(T, \mathsf{cu}+\mathsf{o})$ *will find a solution to it.*

Proof sketch. Assume T has a solution π of length u. Lemma 2 is then used to prove that the search tree constructed by BOUNDEDSEARCH$(T, \mathbf{b})$ will contain $path(\pi)$ when $\mathbf{b} \geq \mathsf{cu}+\mathsf{o}$. When the last node of the path is reached, Lemma 2 and Proposition 1 ensure that the goal formula holds, so the algorithm finds a solution. $\square$

Theorem 5 shows that BOUNDEDSEARCH will always solve a planning task T if the bound is high enough. Since IBDS calls BOUNDEDSEARCH iteratively with increasing bounds, it will eventually find a solution if one exists.

Theorem 6 (Complexity). BOUNDEDSEARCH(T, b) *runs in time* $|T|^{O(1)} \exp_2^{b+1} O(\mathsf{a}+\mathsf{p})$.[2]

Proof sketch. The algorithm explores a search tree where each node corresponds to a canonically b-contracted state. By induction on b, we show that the number and size of such states are both bounded by $\exp_2^{b+1} O(\mathsf{a}+\mathsf{p})$. For each node, at most $O(|T|)$ updates, b-contractions and formulas checks are performed, each taking time polynomial in $|T|$ and the size of the state. $\square$

The theorem shows that computing epistemic plans is tractable when we put a fixed bound on the reasoning depth and the numbers of agents and atoms ($\mathbf{b}$, a and p). This might be a quite realistic assumptions for many practical applications. The theorem generalises several existing results. If $\mathsf{c} = 0$, then BOUNDEDSEARCH(T, o) will find a solution to T (Theorem 5), and it will do so in time $|T|^{O(1)} \exp_2^{\mathsf{o}+1} O(\mathsf{a}+\mathsf{p})$ (Theorem 6). This shows that we can solve propositional planning tasks (*i.e.*, tasks where all pre- and postconditions are propositional) in $(md(\phi_g)+1)$-EXPTIME, hence providing a new proof of an existing result (Yu, Wen, and Liu 2013; Maubert 2014). Also, if $\mathbf{b} = \mathsf{cu}+\mathsf{o}$, then BOUNDEDSEARCH$(T, \mathbf{b})$ is complete (Theorem 5), and it runs in time $|T|^{O(1)} \exp_2^{\mathbf{b}+1} O(\mathsf{a}+\mathsf{p})$ (Theorem 6). This proves that the plan existence problem in epistemic planning is fixed-parameter tractable (FPT) with parameters a, c, o, p and u, generalizing the existing FPT result for epistemic plan verification (Bolander and Lequen 2022). As the aforementioned paper shows that plan verification is *not* FPT for any subset of these parameters, this is the strongest FPT result for the plan existence problem that one can get (with the given parameter set).

[2]Where the iterated exponential $\exp_a^n x$ is defined as follows: $\exp_a^0 x = x$ and $\exp_a^{n+1} x = a^{(\exp_a^n x)}$.

6 Experimental Evaluation

We have compared MIXED-IBDS against three baseline configurations of IBDS: 1. APPROX-IBDS; 2. EXACT-IBDS (a version that always runs in exact mode by choosing a sufficiently large initial search bound); and 3. ROOTED-IBDS (a variant of MIXED-IBDS where canonical b-contractions are replaced by rooted b-contractions, cf. Definition 9).[3] The algorithms were implemented in C++17 in the novel epistemic planner DAEDALUS (*DynAmic Epistemic and DoxAstic Logic Universal Solver*).[4] We tested each configuration on 406 instances (20 minutes timeout) from 9 epistemic planning domains (described in the Appendix): *Active Muddy Child*, *Collaboration through Communication*, *Selective Communication* (Kominis and Geffner 2015), *Coin in the Box* (Baral et al. 2015), *Consecutive Numbers* (van Ditmarsch and Kooi 2015), *Gossip* (Attamah et al. 2014; van Ditmarsch 2016; van Ditmarsch et al. 2017), *Grapevine* (Muise et al. 2015) and *Tiger* (Herzig et al. 2000), plus a novel domain called *Eavesdropping*. Recently, Bolander, Dissing, and Herrmann (2021a) developed a DEL solver that constructs *policies* (mappings from states to actions), which are more general than sequential plans considered in this paper, so we can not directly compare the two approaches.

We now analyze the results of MIXED-IBDS against the baseline configurations (Figure 8) and EFP 2.0 (Figure 9).[5]

MIXED vs. EXACT. EXACT-IBDS solved 302 instances (74.4%), while MIXED-IBDS solved 362 (89.2%), $\sim$15% more. Since EXACT-IBDS always runs in exact mode, *i.e.*, no state is approximated by bounded contractions, bounded contractions behave like standard bisimulation contractions. Thus, MIXED-IBDS computes smaller states than EXACT-IBDS, improving runtimes on the vast majority of the cases, (except for Consecutive Numbers, discussed below), and often finding solutions to instances where EXACT-IBDS timed out, as evidenced by the vertical streak of marks in the time plot (Figure 8a). By computing the *average speedup* (the average of the ratios of the running times) of MIXED-IBDS wrt. EXACT-IBDS, we see that the former configuration is 87.7 times faster than the latter, with most significant improvements on Active Muddy Child (211.9$\times$) and Eavesdropping (376.7$\times$), where visited states have a considerable sizes. Specifically, in Eavesdropping the size of states grows exponentially after each action, but since MIXED-IBDS takes 0-contractions, it collapses states to singleton states, while EXACT-IBDS keeps them in full. Finally, in the Consecutive Numbers domain (red star marks), MIXED-IBDS needs $|\pi|$ iterations to compute plan π (cf. Example 6), while EXACT-IBDS only takes one, as the initial search bound is already large enough, and it returns a plan more quickly.

MIXED vs. APPROX. APPROX-IBDS solved 354 instances ($\sim$2% less than MIXED-IBDS). MIXED-IBDS outperforms APPROX-IBDS in Tiger (3.1$\times$), Coin in the Box (3.4$\times$), and Grapevine (74.4$\times$), and is comparable to APPROX-IBDS in the remaining domains (6$\times$ average speedup).

[3]All configurations rely on the same data structures and functions, except for ROOTED-IBDS, which uses rooted b-contractions.

[4]Available at https://github.com/a-burigana/daedalus.

[5]Raw data of tests results are in the Appendix.

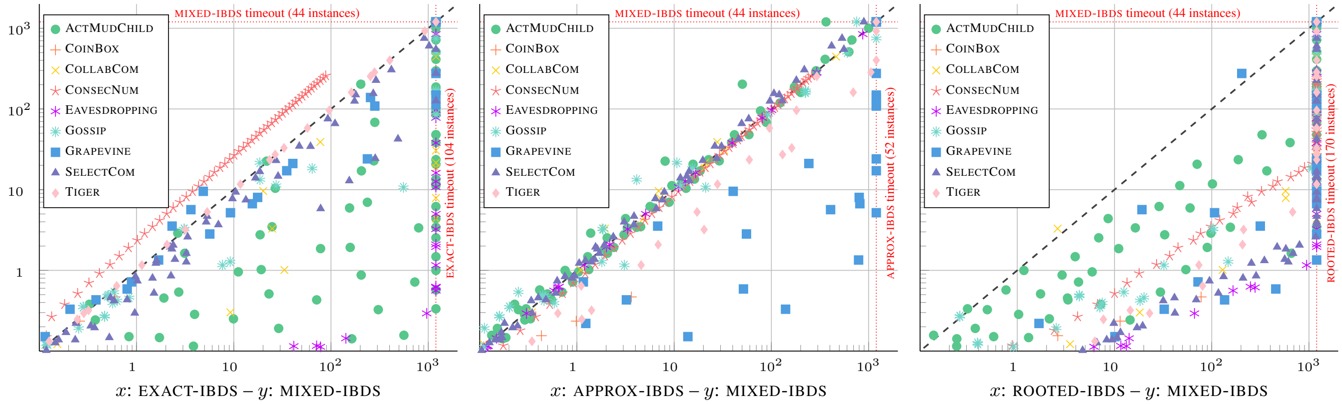

Figure 8: Scatter plots of the running times, in *seconds*, of the algorithms using a logarithmic scale. From left to right, the plots compare the running times of MIXED-IBDS (y-axis) with those of EXACT-IBDS, APPROX-IBDS and ROOTED-IBDS, respectively (x-axis). Color and shape of instances denote their domain. The diagonals (grey dashed lines) separate instances based on which algorithm found a solution faster.

MIXED vs. ROOTED. ROOTED-IBDS solved 236 instances ($\sim$31% less than MIXED-IBDS). The average speedup of MIXED-IBDS wrt. ROOTED-IBDS is 57.2×. As expected, canonical b-contractions allow to efficiently check for visited states via set-membership checks. Instead, with rooted b-contractions, we have to rely on less efficient techniques involving repeated invocations of bounded partition refinement, incurring a significant overhead (cf. Section 4).

MIXED-IBDS vs. EFP 2.0. EFP 2.0 is based on the $m\mathcal{A}^*$ action language (Baral et al. 2015) and uses a *possibility semantics* for DEL, which improves over Kripke-based implementations (Fabiano et al. 2020). Since $m\mathcal{A}^*$ does not have the full expressivity of DEL, we could only compare EFP 2.0 with MIXED-IBDS on a subset of our benchmark tasks (220 tasks in total, excluding Active Muddy Child, Consecutive Numbers, and Eavesdropping). EFP 2.0 solved 95 instances (42.2%), while MIXED-IBDS solved 189 (85.9%). To account for EFP 2.0's overhead in constructing initial states from formulas, we allowed 10 extra minutes per test, but many still timed out. On instances solved by both, MIXED-IBDS was 279.5× faster on average. EFP 2.0 is faster in the Gossip domain (light blue stars above the diagonal in Figure 9), as its possibility semantics enables efficient reuse of previously computed information, especially for private announcements (Fabiano et al. 2020; Burigana, Felli, and Montali 2023), present in Gossip.

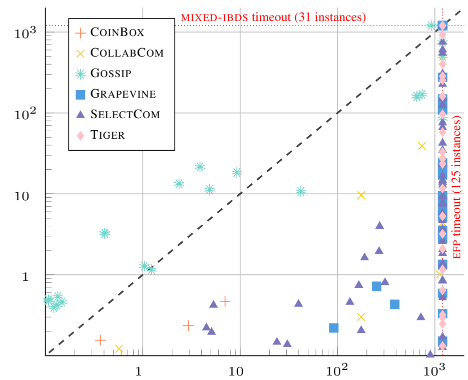

Figure 9: Scatter plot of MIXED-IBDS (y-axis) vs. EFP 2.0 (x-axis).

7 Discussion

Our IBDS algorithm provides a novel perspective on DEL-planning. Instead of searching for plans by iterating on plan length, we iterate on reasoning bound (modal depth). The average modal depth of solutions is 6.9 overall, and 2.4 when excluding the Consecutive Numbers domain that was specifically designed to exploit maximal reasoning depths. This suggests that most planning tasks can be solved with low reasoning depths, an interesting avenue for further exploration. Our novel search strategy improves state-of-the-art (EFP 2.0), showing that fully general DEL planners can be efficient. We conjecture that our approach can also be adapted to outperform the planner of (Bolander, Dissing, and Herrmann 2021b), as their method relies on standard bisimulation contractions only.

Our paper considered only *single-pointed* states, but the *multi-pointed* case can be covered by translating into single-pointed states (Bolander, Dissing, and Herrmann 2021a).

As shown, MIXED-IBDS performs better than EXACT-IBDS in Consecutive Numbers, as it requires more iterations to find a bound that admits a plan. We plan to tackle this issue in the future, e.g. by doubling the bound at each iteration, and/or via heuristics to "guess" a better initial bound.

Acknowledgements

This work is partially funded by the NextGenerationEU FAIR PE0000013 project MAIPM (CUP C63C22000770006). Thomas Bolander is supported by the Independent Research Fund Denmark (grant no. 10.46540/4258-00060B).

References

Attamah, M.; van Ditmarsch, H.; Grossi, D.; and van der Hoek, W. 2014. Knowledge and Gossip. In Schaub, T.; Friedrich, G.; and O'Sullivan, B., eds., *ECAI 2014 - 21st European Conference on Artificial Intelligence, 18-22 August 2014, Prague, Czech Republic - Including Prestigious Applications of Intelligent Systems (PAIS 2014)*, volume 263 of *Frontiers in Artificial Intelligence and Applications*, 21–26. IOS Press.

Aucher, G., and Bolander, T. 2013. Undecidability in epistemic planning. In Rossi, F., ed., *IJCAI 2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence, Beijing, China, August 3-9, 2013*, 27–33. IJCAI/AAAI.

Aucher, G. 2012. DEL-sequents for regression and epistemic planning. *Journal of Applied Non-Classical Logics* 22(4):337–367.

Baltag, A.; Moss, L. S.; and Solecki, S. 1998. The logic of public announcements and common knowledge and private suspicions. In Gilboa, I., ed., *Proceedings of the 7th Conference on Theoretical Aspects of Rationality and Knowledge (TARK-98), Evanston, IL, USA, July 22-24, 1998*, 43–56. Morgan Kaufmann.

Baral, C.; Gelfond, G.; Pontelli, E.; and Son, T. C. 2015. An action language for multi-agent domains: Foundations. *CoRR* abs/1511.01960.

Belle, V.; Bolander, T.; Herzig, A.; and Nebel, B. 2022. Epistemic planning: Perspectives on the special issue. *Artificial Intelligence* 103842.

Blackburn, P.; Rijke, M. d.; and Venema, Y. 2001. *Modal Logic*. Cambridge Tracts in Theoretical Computer Science. Cambridge University Press.

Bolander, T., and Andersen, M. B. 2011. Epistemic planning for single and multi-agent systems. *J. Appl. Non Class. Logics* 21(1):9–34.

Bolander, T., and Burigana, A. 2024. Better bounded bisimulation contractions. In Ciabattoni, A.; Gabelaia, D.; and Sedlár, I., eds., *Advances in Modal Logic, AiML 2024, Prague, Czech Republic, August 19-23, 2024*, 249–268. College Publications.

Bolander, T., and Lequen, A. 2022. Parameterized Complexity of Dynamic Belief Updates: A Complete Map. *Journal of Logic and Computation*.

Bolander, T.; Dissing, L.; and Herrmann, N. 2021a. DEL-based epistemic planning for human-robot collaboration: Theory and implementation. In Bienvenu, M.; Lakemeyer, G.; and Erdem, E., eds., *Proceedings of the 18th International Conference on Principles of Knowledge Representation and Reasoning, KR 2021, Online event, November 3-12, 2021*, 120–129.

Bolander, T.; Dissing, L.; and Herrmann, N. 2021b. DEL-based Epistemic Planning for Human-Robot Collaboration: Theory and Implementation. In *Proceedings of the 18th International Conference on Principles of Knowledge Representation and Reasoning*, 120–129.

Burigana, A.; Felli, P.; Montali, M.; and Troquard, N. 2023. A Semantic Approach to Decidability in Epistemic Planning. In Gal, K.; Nowé, A.; Nalepa, G. J.; Fairstein, R.; and Radulescu, R., eds., *ECAI 2023 - 26th European Conference on Artificial Intelligence, September 30 - October 4, 2023, Kraków, Poland - Including 12th Conference on Prestigious Applications of Intelligent Systems (PAIS 2023)*, volume 372 of *Frontiers in Artificial Intelligence and Applications*, 319–326. IOS Press.

Burigana, A.; Felli, P.; and Montali, M. 2023. DELPHIC: Practical DEL planning via possibilities. In Gaggl, S. A.; Martinez, M. V.; and Ortiz, M., eds., *Logics in Artificial Intelligence - 18th European Conference, JELIA 2023, Dresden, Germany, September 20-22, 2023, Proceedings*, volume 14281 of *Lecture Notes in Computer Science*, 579–594. Springer.

Fabiano, F.; Burigana, A.; Dovier, A.; and Pontelli, E. 2020. EFP 2.0: A multi-agent epistemic solver with multiple e-state representations. In Beck, J. C.; Buffet, O.; Hoffmann, J.; Karpas, E.; and Sohrabi, S., eds., *Proceedings of the Thirtieth International Conference on Automated Planning and Scheduling, Nancy, France, October 26-30, 2020*, 101–109. AAAI Press.

Geffner, H., and Bonet, B. 2013. *A Concise Introduction to Models and Methods for Automated Planning*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers.

Ghallab, M.; Nau, D. S.; and Traverso, P. 2004. *Automated planning - theory and practice*. Elsevier.

Herzig, A.; Lang, J.; Longin, D.; and Polacsek, T. 2000. A Logic for Planning under Partial Observability. In Kautz, H. A., and Porter, B. W., eds., *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on on Innovative Applications of Artificial Intelligence, July 30 - August 3, 2000, Austin, Texas, USA*, 768–773. AAAI Press / The MIT Press.

Kominis, F., and Geffner, H. 2015. Beliefs In Multiagent Planning: From One Agent to Many. In Brafman, R. I.; Domshlak, C.; Haslum, P.; and Zilberstein, S., eds., *Proceedings of the Twenty-Fifth International Conference on Automated Planning and Scheduling, ICAPS 2015, Jerusalem, Israel, June 7-11, 2015*, 147–155. AAAI Press.

Löwe, B.; Pacuit, E.; and Witzel, A. 2011. DEL planning and some tractable cases. In van Ditmarsch, H.; Lang, J.; and Ju, S., eds., *Logic, Rationality, and Interaction - Third International Workshop, LORI 2011, Guangzhou, China, October 10-13, 2011. Proceedings*, volume 6953 of *Lecture Notes in Computer Science*, 179–192. Springer.

Maubert, B. 2014. *Logical foundations of games with imperfect information : uniform strategies. (Fondations logiques des jeux à information imparfaite : stratégies uniformes)*. Ph.D. Dissertation, University of Rennes 1, France.

Muise, C. J.; Belle, V.; Felli, P.; McIlraith, S. A.; Miller, T.; Pearce, A. R.; and Sonenberg, L. 2015. Planning Over Multi-Agent Epistemic States: A Classical Planning Approach. In Bonet, B., and Koenig, S., eds., *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intel-*

ligence, January 25-30, 2015, Austin, Texas, USA, 3327–3334. AAAI Press.

Muise, C.; Belle, V.; Felli, P.; McIlraith, S. A.; Miller, T.; Pearce, A. R.; and Sonenberg, L. 2022. Efficient Multi-Agent Epistemic Planning: Teaching Planners About Nested Belief. *Artificial Intelligence* 302:103605.

Paige, R., and Tarjan, R. E. 1987. Three partition refinement algorithms. *SIAM J. Comput.* 16(6):973–989.

Pardo, P., and Sadrzadeh, M. 2012. Planning in the logics of communication and change. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems - Volume 3*, AAMAS '12, 1231–1232. International Foundation for Autonomous Agents and Multiagent Systems.

Russell, S. J., and Norvig, P. 2010. *Artificial Intelligence—A Modern Approach, Third International Edition*. Pearson Education London.

van de Pol, I.; van Rooij, I.; and Szymanik, J. 2018. Parameterized complexity of Theory of Mind reasoning in dynamic epistemic logic. *Journal of Logic, Language, and Information* 27:255—-294.

van Ditmarsch, H., and Kooi, B. 2015. *One Hundred Prisoners and a Light Bulb*. Cham: Springer International Publishing. 83–94.

van Ditmarsch, H.; van Eijck, J.; Pardo, P.; Ramezanian, R.; and Schwarzentruber, F. 2017. Epistemic Protocols for Dynamic Gossip. *Journal of Applied Logic* 20:1–31.

van Ditmarsch, H. P.; van der Hoek, W.; and Kooi, B. P. 2007. *Dynamic Epistemic Logic*, volume 337. Springer Netherlands.

van Ditmarsch, H. 2016. Epistemic Gossip Protocols. In Ågotnes, T.; Liao, B.; and Wáng, Y. N., eds., *Proceedings of the 1st Chinese Conference on Logic and Argumentation (CLAR 2016), Hangzhou, China, April 2-3, 2016*, volume 1811 of *CEUR Workshop Proceedings*, 1. CEUR-WS.org.

Yu, Q.; Wen, X.; and Liu, Y. 2013. Multi-agent epistemic explanatory diagnosis via reasoning about actions. In Rossi, F., ed., *IJCAI 2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence, Beijing, China, August 3-9, 2013*, 1183–1190. IJCAI/AAAI.