$\mathcal{H}$-EFP: Bridging Efficiency in Multi-Agent Epistemic Planning with Heuristics
Supplementary Documentation

Francesco Fabiano[1], Theoderic Platt[1], Tran Cao Son[1], Enrico Pontelli[1]

[1]Department of Computer Science
New Mexico State University
Las Cruces, NM 88003
{ffabiano,theop,stran,epontell}@nmsu.edu

The following document provides supplementary information for the paper "$\mathcal{H}$-EFP: Bridging Efficiency in Multi-Agent Epistemic Planning with Heuristics" submitted to *The 25th International Conference on Principles and Practice of Multi-Agent Systems* (PRIMA 2024).

In Section A we will introduce some preliminary notions that will help us throughout the presentation. Section B formally introduces the idea of Epistemic Planning Graph and its components, Section B shows the formal demonstration of Theorem 1, and Section **??** illustrates a graphical example of this data structure. Finally, Section E will present the domains used in the experimental setting and Section F all the results from our experimental evaluations.

A Background

A.1 The Action Language $m\mathcal{A}^*$

The action language that will used to evaluate the correctness of our e-PG is $m\mathcal{A}^*$. This action language is an extension of the language m$\mathcal{A}$+ introduced in [1]. $m\mathcal{A}^*$ is a high-level action language for epistemic planning in multi-agent domains. In what follows, we will simply talk about "formulae" instead of "belief formulae", whenever there is no risk of confusion. The notion of a Kripke structure is defined next.

A.1.1 Kripke Structures

To better understand the language $m\mathcal{A}^*$, let us re-introduce the concept of *pointed Kripke Structure*.

Definition 1 (Kripke Structure). *A* Kripke structure*, also referred to as* epistemic model *(*e-model*), is a tuple* $\langle S, \pi, \mathcal{B}_1, \ldots, \mathcal{B}_n \rangle$*, where*

- S *is a set of worlds,*
- $\pi : S \mapsto 2^{\mathcal{F}}$ *is a function that associates an interpretation of* $\mathcal{F}$ *to each element of* S*,*
- *For* $1 \leq i \leq n$*,* $\mathcal{B}_i \subseteq S \times S$ *is a binary relation over* S*.*

Definition 2 (Pointed Kripke Structure). *A* Pointed Kripke structure*, also referred to as* pointed epistemic model *(*pe-model*), is a pair* (M, s) *where* $M = \langle S, \pi, \mathcal{B}_1, \ldots, \mathcal{B}_n \rangle$ *is a Kripke structure and* $s \in S$*. In a pointed Kripke structure* (M, s)*, we refer to* s *as the* real *(or actual)* world.

As in the main paper, for the sake of readability, we use $M[S]$, $M[\pi]$, and $M[i]$ to denote the components S, π, and $\mathcal{B}_i$, of M, respectively. We write $M[\pi](u)$ to denote the interpretation associated to u via π and $M[\pi](u)(\varphi)$ to denote the truth value of a fluent formula φ with respect to the interpretation $M[\pi](u)$.

We will no further describe the properties of the Kripke structures since those are not needed to for the contribution of this work. Moreover, these are extensively defined in [1].

A.1.2 $m\mathcal{A}^*$ Transition Function

$m\mathcal{A}^*$ describes three types of actions that an agent can perform: *world-altering* actions (also known as *ontic* actions), *sensing* actions, and *announcement* actions. Intuitively,

- A **world-altering** action is used to explicitly modify certain properties of the world.
- A **sensing** action is used by an agent to refine its beliefs about the world, by making direct observations.
- An **announcement** action is used by an agent to affect the beliefs of the agents receiving the communication.

In general, as for other types of planning, an action can be executed only when the executability conditions are respected.

Let us now introduce some necessary definitions to present the transition function of $m\mathcal{A}^*$.

Definition 3 ($\mathcal{L}_{\mathcal{AG}}$-substitution). *An* $\mathcal{L}_{\mathcal{AG}}$-substitution *is a set* $\{p_1 \rightarrow \varphi_1, \ldots, p_k \rightarrow \varphi_k\}$, *where each* p_i *is a distinct proposition in* $\mathcal{F}$ *and each* $\varphi_i \in \mathcal{L}_{\mathcal{AG}}$. *We will implicitly assume that for each* $p \in \mathcal{F} \setminus \{p_1, \ldots, p_k\}$, *the substitution contains* $p \rightarrow p$. $SUB_{\mathcal{F},\mathcal{AG}}$ *denotes the set of all* $\mathcal{L}_{\mathcal{AG}}$-*substitutions.*

Definition 4 (Event Model). *Given a set* $\mathcal{AG}$ *of* n *agents, an* event model $\mathbf{\Sigma}$ *is a tuple* $\langle E, Q, pre, sub\rangle$ *where*

- E *is a set, whose elements are called* events*;*
- $Q : \mathcal{AG} \rightarrow 2^{E\times E}$ *assigns an accessibility relation to each agent* $i \in \mathcal{AG}$*;*
- $pre : E \rightarrow \mathcal{L}_{\mathcal{AG}}$ *is a function mapping each event* $e \in E$ *to a formula in* $\mathcal{L}_{\mathcal{AG}}$*; and*
- $sub : E \rightarrow SUB_{\mathcal{F},\mathcal{AG}}$ *is a function mapping each event* $e \in E$ *to a substitution in* $SUB_{\mathcal{F},\mathcal{AG}}$.

Definition 5 ($m\mathcal{A}^*$ Action). *An* epistemic action *is a pair* $(\mathcal{E}, E_d)$, *consisting of an event model* $\mathcal{E} = (E, Q, pre, sub)$ *and a non-empty set of designated events* $E_d \subseteq E$.

Definition 6 ($m\mathcal{A}^*$ Action Execution). *Given an action* $(\mathcal{E}, E_d)$ *and an epistemic state* $(\mathcal{M}, W_d)$, *we say that* $(\mathcal{E}, E_d)$ *is executable in* $(\mathcal{M}, W_d)$ *if, for each* $w \in W_d$, *there exists at least one* $e \in E_d$ *such that* $(\mathcal{M}, w) \models pre(e)$. *The execution of* $(\mathcal{E}, E_d)$ *in* $(\mathcal{M}, W_d)$ *results in an epistemic state* $(\mathcal{M}, W_d) \otimes (\mathcal{E}, E_d) = ((W', \mathcal{R}', \pi'), W'_d)$ *where*

- $W' = \{(w, e) \in W \times E \mid (\mathcal{M}, w) \models pre(e)\}$
- $\mathcal{R}'_i = \{((w, e), (v, f)) \in W' \times W' \mid w\mathcal{R}_i v \wedge eQ_i f\}$
- *For each* $(w, e) \in W'$ *and* $p \in \mathcal{F}$, $\pi'((w, e))(p)$ *is true iff* $p \rightarrow \varphi \in sub(e)$ *and* $(\mathcal{M}, w) \models \varphi$
- $W'_d = \{(w, e) \in W' \mid w \in W_d \, and \, e \in E_d\}$

Let $(\mathcal{M}, w)$ be a pe-model and a be an action executable in $(\mathcal{M}, w)$. The epistemic action corresponding to the occurrence of a in $(\mathcal{M}, w)$ is defined as follows:

Definition 7 (Ontic Action Occurrence). *Let a be an ontic action with the precondition ψ and $(\mathcal{M}, w)$ a pointed epistemic model. The action representing the occurrence of a in $(\mathcal{M}, w)$ is the pair $(\Delta(a, \mathcal{M}, w), \{\sigma\})$ where $\Delta(a, \mathcal{M}, w) = \langle E, Q, pre, sub\rangle$ with*
- $E = \{\sigma, \epsilon\}$ *if* $O_D(a, \mathcal{M}, w) \neq \emptyset$ *and* $E = \{\sigma\}$ *otherwise;*
- $Q(i) = \mathcal{R}_i$ *where*[1]
  - $\mathcal{R}_i = \{(\sigma, \sigma), (\epsilon, \epsilon)\}$ *for* $i \in F_D(a, \mathcal{M}, w)$ *and*
  - $\mathcal{R}_i = \{(\sigma, \epsilon), (\epsilon, \epsilon)\}$ *for* $i \in O_D(a, \mathcal{M}, w)$*;*
- $pre(\sigma) = \psi$ *and* $pre(\epsilon) = \top$*; and*
- $sub(\epsilon) = \emptyset$ *and* $sub(\sigma) = \{p \to \Psi^+(p, a) \vee (p \wedge \neg\Psi^-(p, a)) \mid p \in \mathcal{F}\}$, *where* $\Psi^+(p, a) = \bigvee\{\varphi \mid [\texttt{a}\ \textbf{causes}\ \texttt{p}\ \textit{if}\ \varphi] \in D\}$ *and* $\Psi^-(p, a) = \bigvee\{\varphi \mid [\texttt{a}\ \textbf{causes}\ \neg\texttt{p}\ \textit{if}\ \varphi] \in D\}$.

Definition 8 (Epistemic Action Occurrence). *Let a be a sensing/announcement action with the precondition ψ and the sensed fluent f, and $(\mathcal{M}, w)$ a pointed epistemic model. The epistemic action representing the execution of a in $(\mathcal{M}, w)$ is the pair $(\Delta(a, \mathcal{M}, w), \{\sigma, \tau\})$ where $\Delta(a, \mathcal{M}, w) = \langle E, Q, pre, sub\rangle$ with*
- $E = \{\sigma, \tau, \epsilon\}$*;*
- $Q(i) = \mathcal{R}_i$ *where*
  - $\mathcal{R}_i = \{(\sigma, \sigma), (\tau, \tau), (\epsilon, \epsilon)\}$ *for* $i \in F_D(a, \mathcal{M}, w)$
  - $\mathcal{R}_i = \{(\sigma, \sigma), (\tau, \tau), (\epsilon, \epsilon), (\sigma, \tau), (\tau, \sigma)\}$ *for* $i \in P_D(a, \mathcal{M}, w)$
  - $\mathcal{R}_i = \{(\sigma, \epsilon), (\tau, \epsilon), (\epsilon, \epsilon)\}$ *for* $i \in O_D(a, \mathcal{M}, w)$
- $pre(\sigma) = \psi \wedge f$, $pre(\tau) = \psi \wedge \neg f$, *and* $pre(\epsilon) = \top$*;*
- $sub(x) = \emptyset$ *for each* $x \in E$.

B *e*-PG 2.0: Epistemic Planning Graph

As shown in the previous section, the introduction of heuristics might drastically increase the performance of MEP solvers. To this end, in this article we present e-PG, a Planning Graph that can be used to reason on the full extent of $m\mathcal{A}^*$.

While the definition of a Planning Graph in the classical setting is relatively "easy", the same is not true in the epistemic scenario. In particular, formalizing state-levels for e-PG represents a challenge on its own. In fact, simply putting all the fluent literals in the states would not capture enough information, and using complete e-states would result in a massive overhead, given that they are graph-like structures and their manipulation is very resource-heavy.

The epistemic Planning Graph presented in [7], referred to as e-PG 1.0 from now on, solved this problem by defining each state-level as a set of partial Kripke structures. This allowed capturing enough information without aggravating the e-PG 1.0 resource consumption. Nonetheless, this choice did not allow e-PG 1.0 to work whenever a goal with negated beliefs was requested by the problem.

To overcome such a problem we decided to design a new version of the epistemic Planning Graph, called e-PG 2.0 (Definition 17). In what follows we will formally introduce its main components while providing an intuitive explanation of their behaviour.

First, let us describe what a *state-level* is in e-PG 2.0 (Defintion 11). Following the classical Planning Graph design, we envisioned a state-level to be comprised of a set of formulae, intuitively

[1] When ϵ does not belong to E, the links associated with ϵ are removed.

those that are true at that state-level. Given that the set of possible belief formulae in a MEP problem is infinite, and we need our state-levels to have bounded size to ensure termination, we introduce the idea of *formula of interest*, inspired by [3]. That is, a formula of interest is a belief formula that appears in the domain description, *i.e.*, a single fluent literals, an initial descriptor, a goal, an action precondition, or an observability condition (and its negation to check for oblivious agents). Moreover, to allow for an easier transition function definition, we also consider formulae of interest any *subformula*, as in Definition 9, of the previously mentioned formulae. This set of formulae is formally described in Definition 10.

Definition 9 (Subformulae). *For a belief formula φ, $\mathtt{sf}(\varphi)$ is defined as follows:*

- $\mathtt{sf}(\varphi) = \{\varphi\}$ *if φ is a fluent literal;*
- $\mathtt{sf}(\varphi) = \{\varphi\} \cup \mathtt{sf}(\psi)$ *if $\varphi = \mathbf{B}_{\mathsf{i}}(\psi)$ for some $\mathsf{i} \in \mathcal{AG}$;*
- $\mathtt{sf}(\varphi) = \{\varphi\} \cup \mathtt{sf}(\psi_1) \cup \mathtt{sf}(\psi_2)$ *if $\varphi = \psi_1 \;\mathtt{op}\; \psi_2$ for $\mathtt{op} \in \{\wedge, \vee, \Rightarrow\}$; and*
- $\mathtt{sf}(\varphi) = \{\varphi\} \cup \mathtt{sf}(\psi) \cup \{\mathbf{B}_{\mathsf{i}}(\psi) \mid \mathsf{i} \in \alpha\}$ *if $\varphi = \mathbf{C}_\alpha(\psi)$ where $\alpha \subseteq \mathcal{AG}$.*

Definition 10 (Formulae of interest). *Given a MEP problem $P = \langle\langle \mathcal{F}, \mathcal{AG}, \mathcal{A}\rangle, \mathcal{I}, \mathcal{G}\rangle$, we defined its set of* formulae of interest $\Omega_P = \bigcup_{\psi \in \omega_P} \mathtt{sf}(\psi)$*, assumed to be in CNF, where:*

$$\begin{aligned}\omega_P = &\{\mathtt{f}, \neg\mathtt{f} \mid \mathtt{f} \in P(\mathcal{F})\} \cup \{\varphi_e \mid \varphi_e \in \textit{ exec. conditions of } \mathtt{a}, \mathtt{a} \in P(\mathcal{AI})\} \\ &\cup \{\varphi_o, \neg\varphi_o \mid \varphi_o \in \textit{ obsv. conditions of } \mathtt{a}, \mathtt{a} \in P(\mathcal{AI})\} \\ &\cup \{\varphi_i \mid \varphi_i \in P(\mathcal{I})\} \cup \{\varphi_g \mid \varphi_g \in P(\mathcal{G})\}.\end{aligned}$$

After introducing Ω_P, we can present the notion of a state-level for e-PG 2.0 in Definition 11.

Definition 11 (e-PG 2.0 – State-level). *Given a MEP problem P, a* state-level *of the Planning Graph of P is a set $S \subseteq \Omega_P$ of formulae of interest.*

In Definition 12, we outline how to determine the validity of formulae within a given state-level in e-PG 2.0, to check for action executability, goals, and other conditions. Since, by construction, a state-level represents a collection of true formulae, all formulae within it are inherently considered true. Furthermore, to ensure accurate derivation, especially post-action execution, we permit the derivation of belief formulae by analyzing their subformulae, as shown in Items 2 and 3 of Definition 12. This methodology ensures that e-PG 2.0 provides an overestimation of the planning process.

Definition 12 (e-PG 2.0 – Formulae derivation). *Given a state-level S and a belief formula φ we say that S derives φ, denoted by $S {\mid\!\sim} \varphi$, if:*

- $\varphi \in S$*;*
- *if $\varphi = \psi_1 \;\mathtt{op}\; \psi_2$ and $S {\mid\!\sim} \psi_1 \;\mathtt{op}\; S {\mid\!\sim} \psi_2$ for $\mathtt{op} \in \{\wedge, \vee, \Rightarrow\}$;*
- *if $\varphi = \mathbf{C}_\alpha(\psi)$ and $\forall \mathsf{i} \in \alpha : \mathbf{B}_{\mathsf{i}}(\psi) \in S$, where $\alpha \subseteq \mathcal{AG}$.*

Next, we outline the outcome of executing an *applicable* action (Definition 13) in e-PG 2.0.

Definition 13 (Applicable action instance). *Given a MEP problem P, $S \subseteq \Omega_P$ a state-level in the Planning Graph of P, and an action instance $\mathtt{a} \in P(\mathcal{AI})$, $\mathtt{a}$ is* applicable *in S if $S {\mid\!\sim} \varphi$, where φ is the executability condition of $\mathtt{a}$.*

Defintion 14 presents the direct effects of the action in $m\mathcal{A}^*$, *i.e.*, those properties that after the application of the transition function in $m\mathcal{A}^*$must hold in the updated e-state, as demonstrated in [2].

Definition 14 (Direct effects of actions in $m\mathcal{A}^*$). *Let P be MEP problem, (M, s) a possible e-state of P (indicated as $(M, s) \in P$), $\mathsf{a} \in P(\mathcal{AI})$ an action instance executable in (M, s). Then we define $e_m(\mathsf{a}, (M, s))$, the set of effects of a in (M, s), as follows:*

– *if a is an ontic action and "a **causes** ℓ" in P then*

$$e_m(\mathsf{a}, (M, s)) = \{\ell, \mathbf{B}_\mathsf{i}\ell \mid \mathsf{i} \in \mathbf{F}_\mathsf{a}\} \cup \{\mathbf{B}_\mathsf{j}(\phi), \mathbf{B}_\mathsf{i}\mathbf{B}_\mathsf{j}(\phi) \mid \mathsf{j} \in \mathbf{O}_\mathsf{a}, \mathsf{i} \in \mathbf{F}_\mathsf{a}, (M, s) \models \mathbf{B}_\mathsf{j}(\phi)\}$$

– *if a is sensing action and "a **determines** ψ" in P or an announcement action with "a **announces** ψ" in P then*

$$e_m(\mathsf{a}, (M, s)) = \{\mathbf{C}_{\mathbf{F}_\mathsf{a}}(\psi), \mathbf{C}_{\mathbf{P}_\mathsf{a}}(\mathbf{C}_{\mathbf{F}_\mathsf{a}}(\psi) \lor \mathbf{C}_{\mathbf{F}_\mathsf{a}}(\neg\psi)), \mathbf{C}_{\mathbf{F}_\mathsf{a}}(\mathbf{C}_{\mathbf{P}_\mathsf{a}}(\mathbf{C}_{\mathbf{F}_\mathsf{a}}(\psi) \lor \mathbf{C}_{\mathbf{F}_\mathsf{a}}(\neg\psi)))\} \cup \{\mathbf{B}_\mathsf{j}(\phi), \mathbf{B}_\mathsf{i}\mathbf{B}_\mathsf{j}(\phi) \mid \mathsf{j} \in \mathbf{O}_\mathsf{a}, \mathsf{i} \in \mathbf{F}_\mathsf{a}, (M, s) \models \mathbf{B}_\mathsf{j}(\phi)\}$$

Definition 16 intuitively captures an overestimation of the aforementioned direct effects of $m\mathcal{A}^*$ encompassing every formula of interest that can be derived by these effects with any chains of potentially fully or partially observers (as defined in Definition 15). Essentially, the effects of Definition 16 encapsulate the set of formulae of interest that will be recognized as true post-execution of an action instance a applicable in a state-level S_i. This set of formulae, therefore, represents an overestimation of the action's direct effects following the semantics of $m\mathcal{A}^*$.

Definition 15 (Possible observabilities). *Given a MEP problem P, $S \subseteq \Omega_P$ a state-level in the Planning Graph of P, and an action instance $\mathsf{a} \in P(\mathcal{AI})$ applicable in S, then we identify:*

– $\mathbf{F}^S_\mathsf{a} = \{\mathsf{i} \in P(\mathcal{AG}) \mid$ "i ***observes*** a *if* φ" *in* P, $S \mid\!\sim \varphi\}$
– $\mathbf{P}^S_\mathsf{a} = \{\mathsf{i} \in P(\mathcal{AG}) \mid$ "i ***aware_of*** a *if* φ" *in* P, $S \mid\!\sim \varphi\}$

Definition 16 (e-PG 2.0 – Actions effects). *Let P be a MEP problem, S be a state-level in the Planning Graph $\mathcal{PG}$ of P, a be an action instance $\in P(\mathcal{AI})$ applicable in S, and $\mathbf{F}^S_\mathsf{a}$ and $\mathbf{P}^S_\mathsf{a}$ be the sets of possible observabilities as in Definition 15. We define $e(\mathsf{a}, S)$, the set of effects of a in S in $\mathcal{PG}$, as follows:*

– *if a is an ontic action and "a **causes** ℓ" in P then*
  1. *$\ell \in e(\mathsf{a}, S)$;*
  2. *if $\varphi = \mathbf{C}_\mathsf{X}(\ell)$ belongs to Ω_P and $\mathsf{X} \subseteq \mathbf{F}^S_\mathsf{a}$ then φ belongs to $e(\mathsf{a}, S)$; and*
  3. *if $\varphi = \mathbf{B}_{\mathsf{i}_1} \dots \mathbf{B}_{\mathsf{i}_\mathsf{k}}(\ell)$ belongs to Ω_P and $\{\mathsf{i}_1, \dots, \mathsf{i}_\mathsf{k}\} \subseteq \mathbf{F}^S_\mathsf{a}$ then φ belongs to $e(\mathsf{a}, S)$.*

– *if a is sensing action and "a **determines** ψ" in P or an announcement action with "a **announces** ψ" in P then*
  4. *if $\varphi = \mathbf{B}_{\mathsf{i}_1} \dots \mathbf{B}_{\mathsf{i}_\mathsf{k}}(\psi)$ belongs to Ω_P and $\{\mathsf{i}_1, \dots, \mathsf{i}_\mathsf{k}\} \subseteq \mathbf{F}^S_\mathsf{a}$ then φ belongs to $e(\mathsf{a}, S)$;*
  5. *if $\varphi = \mathbf{C}_\mathsf{X}(\psi)$ belongs to Ω_P and $\mathsf{X} \subseteq \mathbf{F}^S_\mathsf{a}$ then φ belongs to $e(\mathsf{a}, S)$; and*
  6. *if $\varphi = \mathbf{B}_{\mathsf{i}_1} \dots \mathbf{B}_{\mathsf{i}_\mathsf{k}}(\mathbf{C}_\mathsf{X}(\psi) \lor \mathbf{C}_\mathsf{X}(\neg\psi))$ belongs to Ω_P, $\{\mathsf{i}_1, \dots, \mathsf{i}_\mathsf{k}\} \subseteq \mathbf{F}^S_\mathsf{a} \cup \mathbf{P}^S_\mathsf{a}$, and $\mathsf{X} \subseteq \mathbf{F}^S_\mathsf{a}$ then φ belongs to $e(\mathsf{a}, S)$.*

7. *if* $\varphi = \{\mathbf{C}_{\mathsf{Z}}(\mathbf{C}_{\mathsf{Y}}(\mathbf{C}_{\mathsf{X}}(\psi) \vee \mathbf{C}_{\mathsf{X}}(\neg\psi)))$ *belongs to* Ω_P, $\mathsf{Z} \subseteq \mathbf{F}^S_{\mathsf{a}}$, $\mathsf{Y} \subseteq \mathbf{P}^S_{\mathsf{a}}$, *and* $\mathsf{X} \subseteq \mathbf{F}^S_{\mathsf{a}}$ *then* φ *belongs to* $e(\mathsf{a}, S)$.

After defining all the essential components, Definition 17 delineates the computation process for constructing e-PG 2.0. Intuitively, this process unfolds as follows:

- First of all, we build the initial state-level, *i.e.*, S_0, that will contain all the formulae of interest that are entailed by the initial state of the planning problem.
- After that, we iteratively execute the following procedure until the goal is satisfied by one of the state-levels or we reach a fixed point[2]:
  - We check if the state-level derives all the goal conditions. If it does, we found the goal.
  - If the goal is not found we then execute all the executable actions on the state-level producing a new one.
  - If the new state differs, *i.e.*, has some new verified formulae, we reiterate the procedure, otherwise we reach the fixed point concluding that the problem cannot be solved.

Definition 17 (e-PG 2.0). *Given a MEP problem P, its initial e-state $\mathcal{I}$; the* Planning Graph *of P is an alternate sequence of state-levels (as in Definition 11) and action-levels $S_0, A_0, \ldots, S_k, A_k, \ldots$ where:*

- $S_0 = \{\varphi \mid \varphi \in \Omega_P, \mathcal{I} \models \varphi\} \cup \{\psi_1, \psi_2 \mid \varphi = \psi_1 \vee \psi_2 \in \Omega_P, \mathcal{I} \models \varphi\}$;
- *for* $i \geq 0$,
  - A_i *is the set of action instances applicable in* S_i*; and*
  - $S_{i+1} = S_i \cup (\bigcup_{a \in A_i} e(\mathsf{a}, S_i))$

Finally, after constructing the Planning Graph, we can extrapolate useful information to build diverse heuristics. We will explore these in the next section. Furthermore, we show in Theorem 1 that e-PG 2.0 provides a permissible simplification of the actual plan with respect to the semantics of $m\mathcal{A}^*$. Observe that the planning graph will eventually level off because of the finiteness of Ω_P. The proof for Theorem 1 is presented in Appendix C, while a simple example of e-PG 2.0 is shown in Appendix D.

Theorem 1 (e-PG 2.0 Correctness). *Let P be a planning problem, $\mathcal{PG}$ its Planning Graph, and $\varphi \in \Omega_P$ a formula of interest such that $\mathcal{PG}(S_n) \mid\!\sim \varphi$ and $\mathcal{PG}(S_k) \not\mid\!\sim \varphi$ for $k = 0, \ldots, n-1$. Since for all $\mathsf{a} \in P(\mathcal{AI})$ and $(M, s) \in P$ we have that $\{\psi \mid (M, s) \not\models \psi, \Phi_D(\mathsf{a}, (M, s)) \models \psi\} \subseteq e_m(\mathsf{a}, (M, s))$, then the shortest plan to achieve φ is at least of length n.*

Let us note that e-PG 2.0 does not account for indirect effects as potential outcomes of action application as we assume the results of the action to be fully described by its direct effects as presented in Definition 14, expressed, in Theorem 1 as "for all $\mathsf{a} \in P(\mathcal{AI})$ and $(M, s) \in P$ we have that $\{\psi \mid (M, s) \not\models \psi, \Phi_D(\mathsf{a}, (M, s)) \models \psi\} \subseteq e_m(\mathsf{a}, (M, s))$". Since these effects are not integral to the semantics of $m\mathcal{A}^*$ and may not be crucial for Planning Graph construction, their exploration is reserved for future works.

[2] A fixed point is reached whenever a state-level and its successor are identical.

C *e*-PG 2.0 Correctness Demonstration

Theorem 1 (*e*-PG 2.0 Correctness). *Let P be a planning problem, $\mathcal{PG}$ its Planning Graph, and $\varphi \in \Omega_P$ a formula of interest such that $\mathcal{PG}(S_n) \mid\!\sim \varphi$ and $\mathcal{PG}(S_k) \not\mid\!\sim \varphi$ for $k = 0, \ldots, n-1$. Since for all $\mathsf{a} \in P(\mathcal{AI})$ and $(M, s) \in P$ we have that $\{\psi \mid (M, s) \not\models \psi, \Phi_D(\mathsf{a}, (M, s)) \models \psi\} \subseteq e_m(\mathsf{a}, (M, s))$, then the shortest plan to achieve φ is at least of length n.*

Proof. We will demonstrate Theorem 1 using induction on the number n of the state-level in $\mathcal{PG}$ needed to reach φ. We also assume that all the actions are executable, the case in which they are not is trivially demonstrated.
Base Case: Let us start by showing that if the initial state-level $\mathcal{PG}(S_0) \mid\!\sim \varphi$ then the theorem is trivially demonstrated (as there is no shorter plan than the one of length 0).
Induction Hypothesis: We assume that all the relevant formulae entailed by $\mathcal{PG}(S_{n-1})$ are reachable in the planning process with plans of length $\geq n-1$.
Inductive Step: Let us now analyze the case when $\mathcal{PG}(S_{n-1}) \not\mid\!\sim \varphi$ and $\mathcal{PG}(S_n) \mid\!\sim \varphi$. Here we need to show that no plan of length $< n$ that entails φ exists.

Assuming that $[\mathsf{a_0}, \ldots, \mathsf{a_k}]$ is an optimal plan for φ where $k \geq 0$, and we will write (M_i, s_i) to denote $\Phi_D(\mathsf{a_i}, \ldots, (\Phi_D(\mathsf{a_1}, P(\mathcal{I}))))$. We want to show that $k \geq n$. We will show that if $(M_i, s_i) \models \psi$ and $\psi \in \Omega_P$ then $\mathcal{PG}(S_i) \mid\!\sim \psi$.

- **Base Case:** For $k = 0$. This means that $\mathsf{a_0}$ is applicable in $(M_0, P(\mathcal{I}))$. This implies that $pre(\mathsf{a_0})$—the precondition of $\mathsf{a_0}$—is satisfied by $P(\mathcal{I})$. By construction of $\mathcal{PG}$, we have that $pre(\mathsf{a_0}) \in S_0$ and thus $\mathsf{a_0}$ is applicable in $\mathcal{PG}(S_0)$. This means that $e(\mathsf{a_0}, S_0) \subseteq S_1$. Assume that $(M_1, s_1) \models \psi$. If $(M_0, s_0) \models \psi$ then we have $\mathcal{PG}(S_0) \mid\!\sim \psi$ and by construction of $\mathcal{PG}$, $\mathcal{PG}(S_1) \mid\!\sim \psi$. If $(M_0, s_0) \not\models \psi$ then by the assumption of the theorem with respect to $\mathsf{a_0}$ and (M_0, s_0), we have that $\{\psi \mid (M_0, s_0) \not\models \psi, (M_1, s_1) = \Phi_D(\mathsf{a_0}, (M_0, s_0)) \models \psi\} \subseteq e_m(\mathsf{a_0}, (M_0, s_0))$, i.e., $\psi \in e_m(\mathsf{a_0}, (M_0, s_0))$. By Definitions 14 and 16, we have that $\psi \in e(\mathsf{a_0}, S_0)$, and therefore, $\mathcal{PG}(S_1) \mid\!\sim \psi$. This is true because the set of effects defined by Definitions 14 is completely contained by the effects defined by Definition 16.
- **Induction Hypothesis:** We assume that if $(M_i, s_i) \models \psi$ and $\psi \in \Omega_P$ then $\mathcal{PG}(S_i) \mid\!\sim \psi$ for $i < n$.
- **Inductive step:** Similar to the base case, omitted for brevity.

Because $[\mathsf{a_0}, \ldots, \mathsf{a_k}]$ is an optimal plan for φ, we have $(M_{k+1}, s_{k+1}) \models \varphi$. The above shows that $\mathcal{PG}(S_{k+1}) \mid\!\sim \varphi$ and therefore, if $k < n-1$, then it contradicts the assumption of the theorem.

D Example of e-PG 2.0 construction

Let us now introduce an example to clarify how e-PG 2.0 can be constructed starting from a MEP problem P_{ex}. To do so we will first present P_{ex}, in Example 1, from the well-known Coin in the Box Domain [2, 7].

Example 1 (P_{ex}: a Coin in the Box problem). Three agents A, B, and C are in a room. In the middle of the room, there is a box containing a coin. It is common knowledge that:

- Nobody knows which face of the coin is showing (indicated by `tails` or ¬`tails`);
- The box is locked (indicated by ¬`opened`) and only agent A can `open` it;
- A can `peek` into the box (if open) to learn the coin status;
- An agent, observing another agent `peeking` into the box, will conclude that the agent who peeked knows the coin status—but without knowing it herself;
- `Distracting` an agent i causes i to not look at the box;
- `Signaling` an agent i causes the agent to look at the box;
- `Announcing` that the coin is showing heads or tails will cause everyone to know this fact; and
- A and C are looking, while B is not looking at the box.

We assume, for simplicity, that the coin lies `tails` up and that only A can `peek`, `distract`, and `signal`. Agent A wishes to know the status of the coin, and she would like agent B to become aware of the fact that A knows the state of the coin, while keeping C in the dark.

It is easy to see that agent A could achieve such goal by: 1. `distracting` C, keeping her from looking at the box; 2. `signaling` B to look at the box; 3. `opening` the box; and 4. `peeking` into the box.

Let us start by presenting the set of formulae $\Omega_{P_{\text{ex}}}$ of interest of P_{ex}, as defined in Definition 10.

$$
\begin{aligned}
\Omega_{P_{\text{ex}}} = \{&\texttt{tails}, \neg\texttt{tails}, \neg\texttt{opened}, && \text{(Fluents)}\\
&\texttt{look\_i}, \neg\texttt{look\_i}, && \text{(Fluent and obs. cond.)}\\
&\texttt{opened} && \text{(Fluent and } \texttt{peek}\langle\mathsf{A}\rangle \text{ exec. cond.)}\\
&\mathbf{C}_{\mathcal{AG}}(\neg\texttt{opened}), \mathbf{C}_{\mathcal{AG}}(\texttt{look\_A}), && \text{(Init. cond.)}\\
&\mathbf{C}_{\mathcal{AG}}(\neg\mathbf{B}_{\mathsf{i}}(\texttt{tails}) \wedge \neg\mathbf{B}_{\mathsf{i}}(\neg\texttt{tails})), && \text{(Init. cond.)}\\
&\mathbf{C}_{\mathcal{AG}}(\neg\texttt{look\_B}), \mathbf{C}_{\mathcal{AG}}(\texttt{look\_C}), && \text{(Init. cond.)}\\
&\mathbf{B}_{\mathsf{A}}(\texttt{tails}), && (\texttt{shout}\langle\mathsf{A}\rangle \text{ exec. and Goal cond.)}\\
&\mathbf{B}_{\mathsf{B}}(\mathbf{B}_{\mathsf{A}}(\texttt{tails}) \vee \mathbf{B}_{\mathsf{A}}(\neg\texttt{tails})), && \text{(Goal cond.)}\\
&\mathbf{B}_{\mathsf{C}}(\neg\mathbf{B}_{\mathsf{A}}(\texttt{tails}) \wedge \neg\mathbf{B}_{\mathsf{A}}(\neg\texttt{tails}))\} && \text{(Goal cond.)}
\end{aligned}
$$

Where $\mathsf{i} \in \mathcal{AG} = \{\mathsf{A}, \mathsf{B}, \mathsf{C}\}$. Let us note that this set is extrapolated by the domain description in $m\mathcal{A}^*$and, for the sake of brevity, we only present a subset of all the formulae of interest, the ones needed to create our example. Also we will show only the actions executable by A for the sake of conciseness, but the planning graph would contain the same action executed by the other agents, *i.e.*, B and C.

We can now, in Figure 1 illustrate how the Planning Graph is constructed for such a planning process. With a slight abuse of notations, we will include in the various state-levels also the formulae that are not verified and we will use the set $\{\varphi \mid I_{\mathcal{PG}} \mid\!\sim \varphi\}$ to indicate all of those formulae verified by the state S_0 keeping explicit those that are goal formulae for clarity.

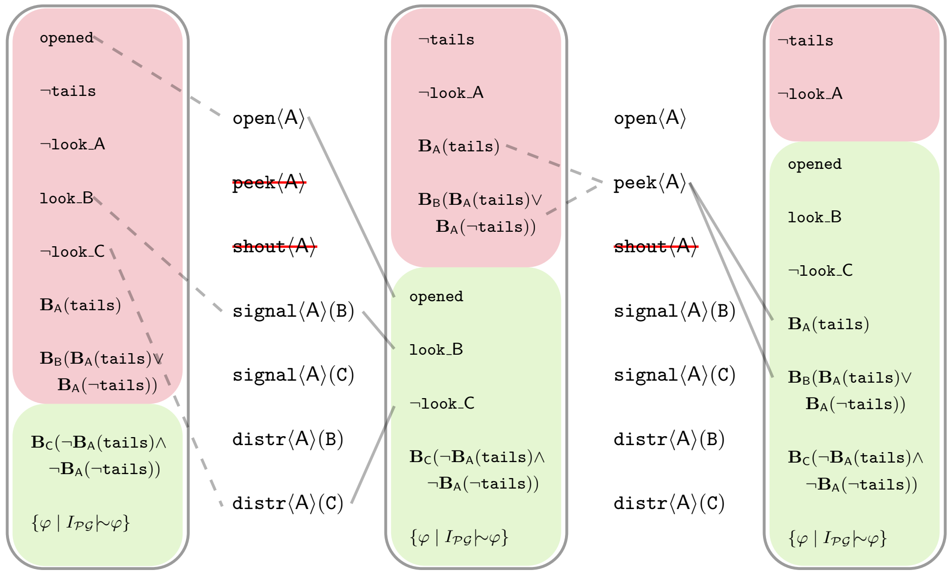

Figure 1: The e-PG 2.0 of P_{ex}. From left to right we have the levels S_0, A_0, S_1, A_2, S_3. Within each state-level, derivable formulae (Definition 11), are enclosed in the lower green box. Additionally, formulae from $\Omega_{P_{\text{ex}}}$ not yet verified are displayed in the upper red box for clarity. We can see that S_3 verifies all goal conditions outlined in $\Omega_{P_{\text{ex}}}$, terminating the construction of e-PG 2.0. An action crossed out in the action-level indicates it is not applicable. A dashed line connecting a formula to an action indicates the formula as a possible effect of that action that will be derived in the subsequent state-level, denoted by a solid line from action-level to state-level.

E Domains Description

In this section, we provide an in-detail description of the benchmarks collected from the literature [4–7] used to test $\mathcal{H}$-EFP.

- *Assembly Line* (**AL**). In this problem, there are two agents, each responsible for processing a different part of a product. Each agent can fail in processing her part and can inform the other agent of the status of her task (action `tell`). Two agents decide to *assemble* (action `act_assemble`) the product or *restart* (action `act_res`), depending on their knowledge about the product status. The goal in this domain is fixed, *i.e.*, the agents must assemble the product, but what varies is the *depth* of the belief formulae used as executability conditions.

- *Coin in the Box* (**CB**). $n \geq 3$ agents are in a room where in the middle there is a box containing a coin. None of the agents know whether the coin lies heads or tails up and the box is locked. One agent has the key to open the box (action `open`). Only *attentive* agents may be aware of the execution of an action. If an agent is attentive, she may look inside the box (action `peek`) to sense the state of the coin. An agent may also share the result (action `shout`). The goals usually consist in some agents knowing whether the coin lies heads or tails up while other agents know that she knows or is ignorant about this.

- *Collaboration and Communication* (**CC**). In this domain, $n \geq 2$ agents move along a corridor with $k \geq 2$ rooms in which $m \geq 1$ boxes can be located. Whenever an agent enters a room, she can determine if a certain box is in the room. Moreover, agents can communicate information about the boxes' position to the other *attentive* agents. Initially, we place all the agents inside room 2. The position of the boxes in initially unknown to each agent. An agent ag may move only to adjacent rooms (actions $\mathsf{left}\langle\mathsf{ag}\rangle$ and $\mathsf{right}\langle\mathsf{ag}\rangle$). To verify the presence of a box b and to communicate it to other agents, an agent can perform the actions $\mathsf{check}\langle\mathsf{ag}\rangle(\mathsf{b})$ and $\mathsf{tell}\langle\mathsf{ag}\rangle(\mathsf{b}, \mathsf{ag}_2)$, respectively.

- *Grapevine* (**Gr**). $n \geq 2$ agents are located in $k \geq 2$ rooms. Each agent ag knows a "secret", represented by the fluent s_ag. An agent can move freely to an adjacent room (actions $\mathsf{left}\langle\mathsf{ag}\rangle$ and $\mathsf{right}\langle\mathsf{ag}\rangle$) and she can share a secret with the agents (action $\mathsf{share}\langle\mathsf{ag}\rangle(\mathsf{s})$) that are in the room with her. This domain supports different goals, from sharing secrets with other agents to having misconceptions about agents' beliefs.

- *Selective Communication* (**SC**). **SC** has $n \geq 2$ agents that start in one of the $k \geq 2$ rooms in a corridor. Every agent is free to move from one room to its adjacent (actions `left` and `right`). In only one of the rooms, an agent may acquire some information, represented by the fluent `q`, by performing the action `sense`. Once an agent acquires such information, she may communicate it to others with the action `shout`. Depending on the room in which this action is performed, different agents *observe* the action. The goals usually require some agents to know certain properties while other agents ignore them.

F Experimental Evaluation

In this section, all the results of the comparison of EFP and $\mathcal{H}$-EFP. For each domain we will present all the results for every execution method, noting that we will use the following acronyms to indicate the diverse resolution strategies:

- **BFS**: Breadth-First Search. This search method is the one used by EFP and therefore we will use it to capture the performance of that solver.
- `C_PG`: Best-First Search where the heuristics emulates the one presented in [7], inspired in turn by the classical way of employing a planning graph. *e*-PG is used to derive the "importance" of each belief formula (its distance from the goal level) and then each e-state is characterized by the sum of the entailed belief formulae scores.
- `L_PG`: Best-First Search where the heuristic calculates the score of an e-state by constructing a planning graph from it (as initial state) and calculating the length—the shorter the better—of the constructed *e*-PG. If an *e*-PG cannot reach the goal from an e-state, then the e-state is discarded.
- `S_PG`: Best-First Search where the heuristics is an execution of `C_PG` on every e-state.
- `SUB`: Best-First Search where heuristics, not dependent from *e*-PG 2.0, simply associates a higher evaluation to e-states that satisfy more sub-goals. To improve this heuristic we defined functions that allow "to break" complex goals into a conjunction of simpler ones to better distinguish between e-states.

Let us note that we will compare the solving times, the number of explored nodes[3], and the length of the found plan.

Moreover, in the following tables, for the sake of readability, we will also make use of these abbreviations:

- 'TO" to indicate Time-Out (solving time over 120 seconds);
- $|\mathcal{AG}|$ to represent the number of agents in the domain;
- $|\mathcal{F}|$ to specify the number of fluents in the domain;
- $|\mathcal{A}|$ to indicate the number of actions in the domain;
- L to point out the optimal length of the plan; and
- d to specify the depth of the belief formulae used in the **Gr** domain. All the plans in this table will have an optimal length of 5 and the use of **C** means that the instance replaces nested belief formulae with ones that employ common knowledge.

Assembly Line																		
$\|\mathcal{AG}\|$	$\|\mathcal{F}\|$	$\|\mathcal{A}\|$	d	Time (seconds)					Expanded Nodes					Plan Length				
				BFS	L_PG	S_PG	C_PG	SUB	**BFS**	L_PG	S_PG	C_PG	SUB	**BFS**	L_PG	S_PG	C_PG	SUB
2	4	6	2	0.02	0.032	0.031	**0.011**	0.018	14	8	8	**5**	10	**5**	**5**	**5**	**5**	**5**
			3	0.021	0.038	0.037	**0.014**	0.019	14	8	8	**5**	10	**5**	**5**	**5**	**5**	**5**
			4	0.029	0.06	0.06	0.034	**0.021**	14	8	8	**5**	10	**5**	**5**	**5**	**5**	**5**
			5	0.121	0.309	0.312	0.207	**0.064**	14	8	8	**5**	10	**5**	**5**	**5**	**5**	**5**
			6	0.588	1.602	1.603	1.374	**0.196**	14	8	8	**5**	10	**5**	**5**	**5**	**5**	**5**
			7	6.438	17.116	17.079	11.907	**2.768**	14	8	8	**5**	10	**5**	**5**	**5**	**5**	**5**
			8	36.671	97.306	97.317	84.478	**11.043**	14	8	8	**5**	10	**5**	**5**	**5**	**5**	**5**
			9	TO	TO	TO	TO	TO	-	-	-	-	-	-	-	-	-	-
			C	0.022	0.039	0.039	**0.015**	0.02	14	8	8	**5**	10	**5**	**5**	**5**	**5**	**5**

Table 1: Performances on the Assembly Line (**AL**) domain of the various solving approaches of $\mathcal{H}$-EFP and EFP.

[3] This measure is important to show that, even if the heuristics implementation is yet not optimized, its ability to reduce the search-space has strong results.

Assembly Line									
$\|\mathcal{AG}\|$	$\|\mathcal{F}\|$	$\|\mathcal{A}\|$	d	Time (seconds)		Expanded Nodes		Plan Length	
				EFP	$\mathcal{H}$-EFP	EFP	$\mathcal{H}$-EFP	EFP	$\mathcal{H}$-EFP
2	4	6	2	0.02	**0.011**	14	**5**	**5**	**5**
			3	0.019	**0.014**	14	**5**	**5**	**5**
			4	0.031	**0.021**	14	**10**	**5**	**5**
			5	0.241	**0.064**	14	**10**	**5**	**5**
			6	0.564	**0.196**	14	**10**	**5**	**5**
			7	6.548	**2.768**	14	**10**	**5**	**5**
			8	35.673	**11.043**	14	**10**	**5**	**5**
			9	TO	TO	-	-	-	-
			C	0.022	**0.015**	14	**5**	**5**	**5**

Table 2: Direct comparison of EFP and $\mathcal{H}$-EFP on the Assembly Line (**AL**) domain.

Coin in the Box																		
$\|\mathcal{AG}\|$	$\|\mathcal{F}\|$	$\|\mathcal{A}\|$	L	Time (seconds)					Expanded Nodes					Plan Length				
				BFS	L_PG	S_PG	C_PG	SUB	**BFS**	L_PG	S_PG	C_PG	SUB	**BFS**	L_PG	S_PG	C_PG	SUB
3	8	21	2	**0.001**	0.023	0.023	0.01	0.001	**2**	**2**	**2**	4	**2**	**2**	**2**	**2**	4	**2**
			2	**0.001**	0.005	0.005	0.001	0.001	**2**	**2**	**2**	**2**	**2**	**2**	**2**	**2**	**2**	**2**
			3	**0.007**	0.101	0.1	0.011	0.013	11	**4**	**4**	**4**	12	**3**	4	4	4	4
			5	0.105	0.372	0.172	0.018	**0.009**	107	10	**5**	**5**	9	**5**	**5**	**5**	**5**	7
			6	1.013	0.662	**0.378**	TO	TO	931	16	**9**	-	-	**6**	7	7	-	-
			7	2.76	0.618	0.299	TO	**0.019**	2585	16	**8**	-	16	**7**	**7**	8	-	8

Table 3: Performances on the Coin in the Box (**CB**) domain of the various solving approaches of $\mathcal{H}$-EFP and EFP.

Coin in the Box									
$\|\mathcal{AG}\|$	$\|\mathcal{F}\|$	$\|\mathcal{A}\|$	L	Time (seconds)		Expanded Nodes		Plan Length	
				EFP	$\mathcal{H}$-EFP	EFP	$\mathcal{H}$-EFP	EFP	$\mathcal{H}$-EFP
3	8	21	2	**0.001**	**0.001**	**2**	**2**	**2**	**2**
			2	**0.001**	0.002	**2**	**2**	**2**	**2**
			3	0.006	**0.005**	**4**	**4**	**3**	**3**
			5	0.123	**0.009**	107	**5**	**5**	7
			6	1.006	**0.378**	931	**9**	**6**	7
			7	2.643	**0.019**	2585	16	**7**	8

Table 4: Direct comparison of EFP and $\mathcal{H}$-EFP on the Coin in the Box (**CB**)) domain.

Collaboration and Communication																		
$\lvert\mathcal{AG}\rvert$	$\lvert\mathcal{F}\rvert$	$\lvert\mathcal{A}\rvert$	L	Time (seconds)					Expanded Nodes					Plan Length				
				BFS	L_PG	S_PG	C_PG	SUB	**BFS**	L_PG	S_PG	C_PG	SUB	**BFS**	L_PG	S_PG	C_PG	SUB
2	10	16	3	0.007	0.028	0.028	0.009	**0.005**	9	**3**	**3**	5	4	**3**	**3**	**3**	5	4
			4	**0.012**	0.022	0.029	TO	0.012	17	**4**	**4**	-	11	**4**	**4**	**4**	-	5
			5	0.066	0.06	0.045	0.007	**0.005**	78	7	**5**	**5**	6	**5**	**5**	**5**	**5**	**5**
			6	0.274	0.083	0.057	TO	**0.02**	301	10	**6**	-	20	**6**	**6**	**6**	-	9
			7	1.125	0.464	0.29	TO	**0.05**	1200	44	**27**	-	50	**7**	12	9	-	11
			8	2.344	**0.214**	TO	TO	TO	2358	**16**	-	-	-	**8**	9	-	-	-
2	14	28	3	0.02	0.055	0.055	0.031	**0.011**	6	**3**	**3**	4	**3**	**3**	**3**	**3**	4	**3**
			4	0.068	0.061	0.066	TO	**0.025**	24	**4**	**4**	-	10	**4**	**4**	**4**	-	5
			5	0.515	0.405	0.11	TO	**0.029**	169	17	**5**	-	7	**5**	8	**5**	-	5
			6	2.512	0.436	0.12	TO	**0.048**	810	19	**6**	-	19	**6**	7	**6**	-	9
			7	12.366	0.957	0.369	TO	**0.088**	4047	39	**14**	-	18	**7**	10	8	-	**7**
2	16	40	3	0.14	0.254	0.253	0.329	**0.084**	6	**3**	**3**	4	**3**	**3**	**3**	**3**	4	**3**
			4	0.649	0.322	0.306	TO	**0.174**	29	**4**	**4**	-	10	**4**	**4**	**4**	-	5
			5	6.493	1.578	0.535	TO	**0.244**	275	18	**5**	-	8	**5**	8	**5**	-	**5**
			6	37.236	2.79	0.396	TO	**0.313**	1611	23	**6**	-	15	**6**	8	**6**	-	7
			7	TO	5.425	2.14	TO	**0.443**	-	49	28	-	**22**	-	12	**9**	-	**9**
2	21	52	3	TO	TO	TO	TO	TO	-	-	-	-	-	-	-	-	-	-
			4	TO	TO	TO	TO	TO	-	-	-	-	-	-	-	-	-	-
			5	TO	TO	TO	TO	TO	-	-	-	-	-	-	-	-	-	-
			6	TO	TO	TO	TO	TO	-	-	-	-	-	-	-	-	-	-
			7	TO	TO	TO	TO	TO	-	-	-	-	-	-	-	-	-	-
3	12	24	3	0.01	0.053	0.053	0.018	**0.005**	8	**3**	**3**	5	**3**	**3**	**3**	**3**	5	**3**
			4	0.038	0.046	0.058	TO	**0.009**	30	**4**	**4**	-	6	**4**	**4**	**4**	-	**4**
			5	0.272	0.119	0.087	0.018	**0.009**	177	7	**5**	6	6	**5**	**5**	**5**	6	**5**
			6	0.771	0.248	0.124	TO	**0.014**	489	13	**6**	-	8	**6**	**6**	**6**	-	7
			7	7.18	1.533	0.747	TO	**0.029**	3962	85	40	-	**13**	**7**	18	12	-	**7**
3	14	42	3	0.036	0.118	0.117	0.095	**0.017**	8	**3**	**3**	5	**3**	**3**	**3**	**3**	5	**3**
			4	0.191	0.171	0.149	0.07	**0.027**	39	5	**4**	5	5	**4**	**4**	**4**	5	**4**
			5	3.302	2.941	0.176	TO	**0.036**	533	72	**5**	-	8	**5**	9	**5**	-	6
			6	9.95	1.149	0.295	TO	**0.036**	1638	28	**8**	-	**8**	**6**	9	**6**	-	**6**
			7	83.486	**23.978**	TO	TO	TO	12093	**499**	-	-	-	**7**	43	-	-	-

Table 5: Performances on the Collaboration and Communication (**CB**) domain of the various solving approaches of $\mathcal{H}$-EFP and EFP.

Collaboration and Communication									
$\|\mathcal{AG}\|$	$\|\mathcal{F}\|$	$\|\mathcal{A}\|$	L	Time (seconds)		Expanded Nodes		Plan Length	
				EFP	$\mathcal{H}$-EFP	EFP	$\mathcal{H}$-EFP	EFP	$\mathcal{H}$-EFP
2	10	16	3	0.007	**0.005**	9	**4**	**3**	4
			4	**0.010**	0.012	**17**	**17**	**4**	**4**
			5	0.072	**0.005**	78	**6**	**5**	**5**
			6	0.267	**0.02**	301	**20**	**6**	9
			7	1.353	**0.05**	1200	**50**	**7**	11
			8	2.144	**0.214**	2358	**16**	**8**	9
2	14	28	3	0.02	**0.011**	6	**3**	**3**	**3**
			4	0.054	**0.025**	24	**10**	**4**	5
			5	0.524	**0.029**	169	**7**	**5**	**5**
			6	2.284	**0.048**	810	**19**	**6**	9
			7	11.874	**0.088**	4047	**18**	**7**	**7**
2	16	40	3	0.231	**0.084**	6	**3**	**3**	**3**
			4	0.562	**0.174**	29	**10**	**4**	5
			5	5.792	**0.244**	275	**8**	**5**	**5**
			6	36.938	**0.313**	1611	15	**6**	7
			7	TO	**0.443**	-	**22**	-	**9**
2	21	52	3	TO	TO	-	-	-	-
			4	TO	TO	-	-	-	-
			5	TO	TO	-	-	-	-
			6	TO	TO	-	-	-	-
			7	TO	TO	-	-	-	-
3	12	24	3	0.01	**0.005**	8	**3**	**3**	**3**
			4	0.034	**0.009**	30	**6**	**4**	**4**
			5	0.263	**0.009**	177	**6**	**5**	**5**
			6	0.837	**0.014**	489	**8**	**6**	7
			7	7.362	**0.029**	3962	**13**	**7**	**7**
3	14	42	3	0.034	**0.017**	8	**3**	**3**	**3**
			4	0.213	**0.027**	39	**5**	**4**	**4**
			5	4.234	**0.036**	533	**8**	**5**	6
			6	10.347	**0.036**	1638	**8**	**6**	**6**
			7	86.247	**23.978**	12093	**499**	**7**	43

Table 6: Direct comparison of EFP and $\mathcal{H}$-EFP on the Collaboration and Communication (**CC**)) domain.

Grapevine																		
$\lvert\mathcal{AG}\rvert$	$\lvert\mathcal{F}\rvert$	$\lvert\mathcal{A}\rvert$	L	Time (seconds)					Expanded Nodes					Plan Length				
				BFS	L_PG	S_PG	C_PG	SUB	**BFS**	L_PG	S_PG	C_PG	SUB	**BFS**	L_PG	S_PG	C_PG	SUB
3	9	24	2	0.009	0.078	0.031	0.008	**0.003**	6	4	**2**	3	**2**	**2**	**2**	**2**	3	**2**
			3	0.058	0.179	0.053	TO	**0.005**	31	8	**3**	-	**3**	**3**	**3**	**3**	-	**3**
			4	0.375	TO	0.064	TO	**0.005**	161	-	**4**	-	**4**	**4**	-	**4**	-	**4**
			5	2.269	15.096	0.128	TO	**0.011**	834	517	**6**	-	**6**	**5**	55	**5**	-	**5**
			6	6.071	TO	0.208	TO	**0.021**	2146	-	**7**	-	**7**	**6**	-	**6**	-	**6**
			7	36.728	TO	**0.91**	TO	TO	12105	-	**26**	-	-	**7**	-	11	-	-
4	12	40	2	0.043	0.204	0.114	0.039	**0.01**	7	3	**2**	3	**2**	**2**	**2**	**2**	3	**2**
			3	0.298	0.766	0.18	TO	**0.017**	40	10	**3**	-	**3**	**3**	**3**	**3**	-	**3**
			4	2.131	3.958	0.222	TO	**0.018**	230	49	**4**	-	**4**	**4**	8	**4**	-	**4**
			5	14.779	TO	TO	TO	**0.072**	1366	-	-	-	**10**	**5**	-	-	-	6
			6	39.722	TO	TO	TO	**0.08**	3706	-	-	-	**7**	**6**	-	-	-	**6**
4	16	60	2	0.342	1.24	0.337	0.244	**0.06**	8	4	**2**	3	**2**	**2**	**2**	**2**	3	**2**
			3	4.065	TO	0.54	TO	**0.093**	71	-	**3**	-	**3**	**3**	-	**3**	-	**3**
			4	46.455	TO	0.566	TO	**0.094**	617	-	**4**	-	**4**	**4**	-	**4**	-	**4**
			5	TO	TO	1.267	TO	**0.236**	-	-	**6**	-	**6**	-	-	**5**	-	**5**
			6	TO	TO	3.325	TO	**1.016**	-	-	**7**	-	11	-	-	**6**	-	7

Table 7: Performances on the Grapevine (**Gr**) domain of the various solving approaches of $\mathcal{H}$-EFP and EFP.

Grapevine									
$\lvert\mathcal{AG}\rvert$	$\lvert\mathcal{F}\rvert$	$\lvert\mathcal{A}\rvert$	L	Time (seconds)		Expanded Nodes		Plan Length	
				EFP	$\mathcal{H}$-EFP	EFP	$\mathcal{H}$-EFP	EFP	$\mathcal{H}$-EFP
3	9	24	2	0.012	**0.003**	6	**2**	**2**	**2**
			3	0.048	**0.005**	31	**3**	**3**	**3**
			4	0.369	**0.005**	161	**4**	**4**	**4**
			5	2.235	**0.011**	834	**6**	**5**	**5**
			6	5.839	**0.021**	2146	**7**	**6**	**6**
			7	37.238	**0.91**	12105	**26**	**7**	11
4	12	40	2	0.047	**0.012**	7	**3**	**2**	**2**
			3	0.352	**0.017**	40	**3**	**3**	**3**
			4	2.253	**0.018**	230	**4**	**4**	**4**
			5	16.384	**0.072**	1366	**10**	**5**	6
			6	38.632	**0.08**	3706	**7**	**6**	**6**
4	16	60	2	0.373	**0.06**	8	**2**	**2**	**2**
			3	4.274	**0.093**	71	**3**	**3**	**3**
			4	43.672	**0.094**	617	**4**	**4**	**4**
			5	TO	**0.236**	-	**6**	-	**5**
			6	TO	**1.016**	-	**11**	-	**7**

Table 8: Direct comparison of EFP and $\mathcal{H}$-EFP on the Grapevine (**Gr**) domain.

Selective Communication																		
$\|\mathcal{AG}\|$	$\|\mathcal{F}\|$	$\|\mathcal{A}\|$	L	Time (seconds)					Expanded Nodes					Plan Length				
				BFS	L_PG	S_PG	C_PG	SUB	**BFS**	L_PG	S_PG	C_PG	SUB	**BFS**	L_PG	S_PG	C_PG	SUB
3	5	7	3	**0.001**	0.006	0.006	**0.001**	**0.001**	4	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**
			5	**0.002**	0.013	0.013	TO	TO	11	**6**	**6**	-	-	**5**	**5**	**5**	-	-
			6	**0.005**	TO	TO	TO	TO	**21**	-	-	-	-	**6**	-	-	-	-
			8	**0.018**	TO	TO	TO	TO	**77**	-	-	-	-	**8**	-	-	-	-
7	5	7	5	**0.007**	0.021	0.021	TO	TO	17	**6**	**6**	-	-	**5**	**5**	**5**	-	-
			7	**0.039**	TO	TO	TO	TO	**89**	-	-	-	-	**7**	-	-	-	-
			8	**0.091**	TO	TO	TO	TO	**208**	-	-	-	-	**8**	-	-	-	-
8	11	13	10	**0.004**	0.046	0.046	TO	0.005	**10**	**10**	**10**	-	11	**10**	**10**	**10**	-	11
			14	**0.062**	0.097	0.095	TO	TO	103	**14**	**14**	-	-	**14**	**14**	**14**	-	-
			15	0.171	0.116	**0.116**	TO	TO	261	**15**	**15**	-	-	**15**	**15**	**15**	-	-
9	11	13	6	0.098	TO	0.063	TO	**0.006**	134	-	**6**	-	**6**	**6**	-	**6**	-	**6**
			8	**0.458**	TO	TO	TO	8.215	**553**	-	-	-	3042	**8**	-	-	-	100
			9	**1.768**	TO	TO	TO	7.959	**1957**	-	-	-	3042	**9**	-	-	-	100
			9	**0.004**	0.04	0.04	0.007	0.005	10	**9**	**9**	**9**	10	**9**	**9**	**9**	**9**	**9**
			10	**0.005**	0.053	0.055	0.009	0.007	12	**10**	**10**	**10**	12	**10**	**10**	**10**	**10**	11
			12	38.353	TO	**0.314**	TO	8.708	37862	-	**14**	-	3295	**12**	-	13	-	130
			13	0.047	TO	0.132	TO	**0.04**	70	-	**14**	-	37	**13**	-	14	-	16
			17	3.576	TO	**0.351**	TO	0.697	4179	-	**22**	-	292	**17**	-	22	-	37
9	12	14	4	**0.003**	0.032	0.028	0.005	**0.003**	6	5	**4**	**4**	**4**	**4**	**4**	**4**	**4**	**4**
			5	0.006	0.055	0.062	0.011	**0.005**	11	8	7	**6**	**6**	**5**	**5**	6	6	6
			6	0.015	0.086	0.084	**0.009**	TO	23	13	13	**6**	-	**6**	**6**	**6**	**6**	-
			7	0.038	0.104	0.096	0.017	**0.008**	55	12	11	**8**	**8**	**7**	**7**	**7**	8	8
			8	0.175	TO	0.134	0.026	**0.012**	133	-	13	**10**	**10**	**8**	-	**8**	10	10
			9	0.262	TO	0.192	0.038	**0.017**	327	-	16	**12**	**12**	**9**	-	10	12	12
			10	0.711	TO	1.6	0.04	**0.019**	829	-	83	**13**	**13**	**10**	-	27	13	13
			11	1.883	TO	0.347	0.052	**0.023**	2152	-	23	**14**	**14**	**11**	-	16	14	14

Table 9: Performances on the Selective Communication (**SC**) domain of the various solving approaches of $\mathcal{H}$-EFP and EFP.

Selective Communication									
$\lvert\mathcal{AG}\rvert$	$\lvert\mathcal{F}\rvert$	$\lvert\mathcal{A}\rvert$	L	Time (seconds)		Expanded Nodes		Plan Length	
				EFP	$\mathcal{H}$-EFP	EFP	$\mathcal{H}$-EFP	EFP	$\mathcal{H}$-EFP
3	5	7	3	**0.001**	**0.001**	4	**3**	**3**	**3**
			5	**0.002**	**0.002**	**11**	**11**	**5**	**5**
			6	**0.004**	0.005	**21**	**21**	**6**	**6**
			8	0.015	**0.013**	**77**	**77**	**8**	**8**
7	5	7	5	**0.007**	**0.007**	**17**	**17**	**5**	**5**
			7	**0.036**	0.039	**89**	**89**	**7**	**7**
			8	0.102	**0.091**	**208**	**208**	**8**	**8**
8	11	13	10	**0.004**	**0.004**	**10**	**10**	**10**	**10**
			14	**0.058**	0.062	**103**	**103**	**14**	**14**
			15	0.185	**0.116**	261	**15**	**15**	**15**
9	11	13	6	0.074	**0.006**	134	**6**	**6**	**6**
			8	0.521	**0.458**	**553**	**553**	**8**	**8**
			9	**1.384**	1.768	**1957**	**1957**	**9**	**9**
			9	0.007	**0.004**	**10**	**10**	**9**	**9**
			10	**0.005**	**0.005**	**12**	**12**	**10**	**10**
			12	36.327	**0.314**	37862	**14**	**12**	13
			13	0.052	**0.04**	70	**37**	**13**	16
			17	3.581	**0.351**	4179	**22**	**17**	22
9	12	14	4	**0.003**	**0.003**	6	**4**	**4**	**4**
			5	0.007	**0.005**	11	**6**	**5**	6
			6	0.023	**0.009**	23	**6**	**6**	**6**
			7	0.034	**0.008**	55	**8**	**7**	8
			8	0.135	**0.012**	133	**10**	**8**	10
			9	0.356	**0.017**	327	**12**	**9**	12
			10	0.726	**0.019**	829	**13**	**10**	13
			11	2.038	**0.023**	2152	**14**	**11**	14

Table 10: Direct comparison of EFP and $\mathcal{H}$-EFP on the Selective Communication (**SC**) domain.

References

1. Chitta Baral, Gregory Gelfond, Enrico Pontelli, and Tran Cao Son. An action language for multi-agent domains: Foundations. *CoRR*, abs/1511.01960, 2015.
2. Chitta Baral, Gregory Gelfond, Enrico Pontelli, and Tran Cao Son. An action language for multi-agent domains. *Artificial Intelligence*, 302:103601, 2022.
3. Alessandro Burigana, Francesco Fabiano, Agostino Dovier, and Enrico Pontelli. Modelling multi-agent epistemic planning in asp. *Theory and Practice of Logic Programming*, 20(5):593–608, 2020.
4. Francesco Fabiano, Alessandro Burigana, Agostino Dovier, and Enrico Pontelli. EFP 2.0: A multi-agent epistemic solver with multiple e-state representations. In *Proceedings ICAPS, Nancy, France, October 26-30, 2020*, pages 101–109. AAAI Press, 2020.
5. Xiao Huang, Biqing Fang, Hai Wan, and Yongmei Liu. A general multi-agent epistemic planner based on higher-order belief change. In *Proceedings of IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, pages 1093–1101. ijcai.org, 2017.
6. Filippos Kominis and Hector Geffner. Beliefs in multiagent planning: From one agent to many. In *Proceedings of the Twenty-Fifth International Conference on Automated Planning and Scheduling, ICAPS 2015, Jerusalem, Israel, June 7-11, 2015*, pages 147–155. AAAI Press, 2015.
7. Tiep Le, Francesco Fabiano, Tran Cao Son, and Enrico Pontelli. EFP and PG-EFP: Epistemic forward search planners in multi-agent domains. In *Proceedings ICAPS*, pages 161–170, Delft, The Netherlands, June 24–29 2018. AAAI Press.