Goal Elicitation Planning: Acting to Reveal the Goals of Others

Adam Amos-Binks AAMOSBINKS@ARA.COM
Applied Research Associates, Raleigh, NC, USA

Rogelio E. Cardona-Rivera ROGELIO@CS.UTAH.EDU
School of Computing and the Entertainment Arts and Engineering Program
University of Utah, Salt Lake City, UT, USA

Abstract

Adapting to and resolving imperfect information is a hallmark of intelligent behavior. Plan, activity, and intention recognition (PAIR) methods recognize agent goals from ambiguous agent behavior so they may predict future action. However, despite the inherent social function of goals to coordinate action, there have been no investigations into goal reasoning agents who adopt PAIR-like methods to resolve and elicit the goals of other agents. Our approach, goal-elicitation planning (GEP), builds atop the goal recognition design (GRD) framework to provide an online mechanism for goal-reasoning agents to adopt goals that, in turn, reveal the goals of another agent. To this end we make three contributions. First, we characterize the properties of a problem that make it conducive to GEP. Second, we define GEP for revealing the goals of optimal (cost) behaving target agents. Our third contribution defines GEP for non-optimal target agents who have a cost budget to deviate from the optimal path towards their goals. Our preliminary results in a modified `easy-grid` benchmark show that GEP achieves online results similar to offline GRD.

1. Introduction

Goal reasoning – the formulation, management, and accomplishment or maintenance of goals – is a hallmark of intelligent behavior (Vattam et al., 2013). In multi-agent contexts, individual agents must resolve incomplete or imperfect goal information to better strategize about cooperative *and* competitive activity (Grosz & Kraus, 1999). Scholars have argued that developing better methods to resolve such goal information will pave the way toward agents with both greater autonomy (e.g., Muñoz-Avila et al., 2019) and higher-order reasoning (e.g., Amos-Binks & Dannenhauer, 2020).

However, despite the inherent social function of goals to coordinate activity (Bratman, 1987; Cohen & Levesque, 1990; Amos-Binks et al., 2019), state-of-the-art goal reasoning agents remain individually limited in their ability to formulate the goals of *others*. While overt communication facilitates such reasoning (Jokinen, 1996), such communication is not always possible or desirable (e.g. in adversarial contexts). The goal formulation methods developed within the plan, activity, and intention recognition (PAIR, Sukthankar et al., 2014) community aim to *recognize* an observed actor's goals from their behavior, in service of predicting the actor's subsequent behavior. Unfortu-

nately, these methods predominantly assume *disembodied recognition*: the observer is never part of the task environment they are observing. As a consequence, the observer can never actively elicit (and must therefore passively recognize) the actor's goal. In other words, goal reasoning agents interested in formulating the goals of *others* cannot yet leverage *acting* within task environments to improve their predictions about what others' goals are. In this paper, we present one way to do precisely that.

Our approach, which we term **goal-elicitation planning** (GEP), is an *online* goal formulation mechanism that enables a goal-reasoning agent to adopt a goal that will in-turn reveal the goal of another agent. A GEP agent – the **elicitor** – must find a sequence of actions that coerces an actor – the **target** – to act in a way that reveals its goal as early as possible.

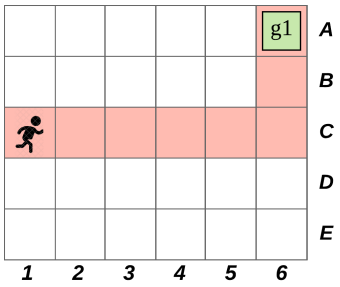
Contributions Our contribution is three-fold. First, we formalize the Goal Elicitation Planning problem, including the necessary conditions of a PAIR task that make it amenable to GEP; we refer to such tasks as goal elicitation *plannable* or *GEPable*. Second, we define the GEP reasoning process for (cost-)optimal agents. Third, we define GEP for bounded non-optimal agents. We present preliminary evaluation results for all goal elicitation methods on the `easy-grid` domain drawn from the PAIR benchmarking literature; the `easy-grid` problem was modified slightly to make it GEPable. Our results indicate that for both optimal and bounded non-optimal agents, the (online) GEP process provides the capability for a goal reasoning agent to generate and execute plans that elicit the target actor's goals as effectively as its (offline) counterpart GRD.

2. Background

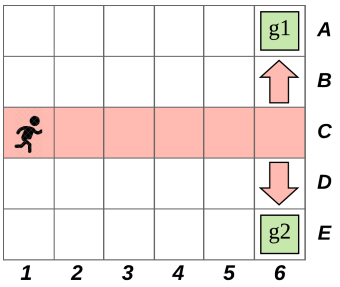
Goal Elicitation Planning depends on several goal-reasoning models. In this section we review each, deferring a more-thorough discussion of related work until we can refer to it more-precisely.

2.1 Classical Planning

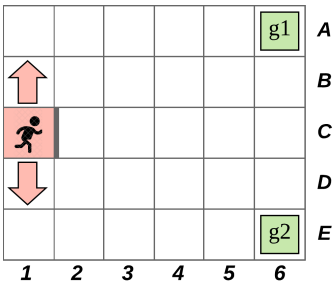
Our work is based on **classical planning**, a problem-solving model wherein agent actions are fully observable and deterministic. We use the STRIPS representation (Fikes & Nilsson, 1971); a STRIPS planning problem is a tuple $P = \langle \mathcal{L}, I, A, G, \gamma, f_{\text{cost}} \rangle$ where $\mathcal{L}$ is the set of literals, $I \subseteq \mathcal{L}$ is an initial state, $G \subseteq \mathcal{L}$ is a set of goal conditions, and A is a set of actions. Each action is a triple $a = \langle \text{PRE}(a), \text{ADD}(a), \text{DEL}(a) \rangle$, that represents the precondition, add, and delete lists respectively, all subsets of $\mathcal{L}$. A state is a set of conjuncted literals, and an action a is applicable in a state s if $\text{PRE}(a) \subseteq s$. Applying said applicable action in the state results in a new state s' as specified by γ, a state-transition function over the set of potential states S; $\gamma \colon S \times A \to 2^S$. The set S is implicitly defined as the powerset of the set of literals $2^{\mathcal{L}}$. For classical planning, given a state s and applicable action a, the resulting new state s' is given by $\gamma(s, a) = (s \setminus \text{DEL}(a)) \cup \text{ADD}(a)$. Applying the applicable action incurs a cost determined by the function $f_{\text{cost}} \colon A \to \mathbb{R}^{>0}$; unless otherwise indicated, we use $f_{\text{cost}} = 1$. The solution to a planning problem P is a plan $\pi = [a_1, ..., a_m]$, a sequence of actions $a_i \in A$ that transforms the problem's initial state I to a state s_m that satisfies the goal; *i.e.* $G \subseteq s_m$. The cost $\text{c}(\pi)$ of a plan π is $\sum_{a_i \in \pi} f_{\text{cost}}(a_i)$. The execution trace $trace(\pi, I) = [I, a_1, s_1, ..., a_m, s_m]$ of plan π from initial state I is an alternating sequence of states



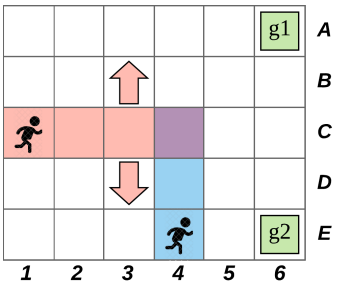
(a) Classical planning: an `Actor` searches for the *cost-optimal* plan to achieve $g1$. In other words, they find the shortest plan – visualized by the red highlighted path – to achieve their goal from a given state.



(b) Goal recognition: an external observer must predict which of the assumed goals $g1$ *or* $g2$ the `Actor` is pursuing, given an unfolding plan. Above, the illustrated plan is the *most ambiguous* possible one.



(c) Goal recognition design: modifying the environment *offline* to minimize worst-case distinctiveness. By removing action `(move Actor C1 C2)` – visualized via a barrier – a designer forces an `Actor` to reveal their goal.



(d) Goal elicitation planning: an *online* variant of GRD. By planning to clobber precondition `(isFree C4)` of action `(move Actor C3 C4)`, the now-embodied observer (*i.e.* the elicitor) in `E4` will thwart the plan in (b).

Figure 1: Goal reasoning methods within the `easy-grid` task environment, comprised of: a two-dimensional matrix of locations (`A1`, . . . , `E6`), adjacency relations between these locations (e.g. `(adj A1 A2)`), and an occupancy attribute over these locations (e.g. `(isFree C1)`). Our work introduces goal elicitation planning, an *online* variant of goal recognition design, in which an elicitor agent – an *embodied* goal recognition observer – finds a plan that minimizes the *wcd* by clobbering the target agent's (i.e. goal recognition actor's) most ambiguous plan at the earliest possible opportunity.

and actions, starting with I, such that s_i results from applying a_i to state s_{i-1}. A solution to a planning problem is found via a search procedure, which must ensure that every action in the plan has its preconditions satisfied (Bonet & Geffner, 2001). Unless specified otherwise, we assume our search aims to find the optimal (*i.e.* lowest cost) plan, which for classical planning is tantamount to the shortest-length plan when action costs are uniform.

```
(:action move   :parameters (?a - agent ?from - loc ?to - loc)
  :precondition (and (at ?a ?from) (adj ?from ?to) (isFree ?to))
  :effect       (and (not (at ?a ?from)) (not (isFree ?to))
                     (at ?a ?to) (isFree ?from)))
```

Listing 1: A `move` operator for an `agent` in the Planning Domain Definition Language (McDermott et al., 1998), requiring the target `loc`ation `isFree` (unoccupied) and `adj`acent.

Classical planning problems are typically codified in the Planning Domain Definition Language (PDDL, McDermott et al., 1998). Listing 1 illustrates a `move` operator, a template for a movement action with parameters, (logical) preconditions, and effects (which combine the add and delete lists for an action). All possible instances of `move` – e.g. `(move Actor C1 C2)`, `(move Actor C2 C3)`, etc.– define the set A and are usable to solve a planning problem in the example `easy-grid` task environment illustrated in Figure 1; Figure 1a illustrates an example plan, which will accomplish an `Actor`'s goal $g1 =$ `(at Actor A6)` from their initial state of being `(at Actor C1)`.

2.2 Goal Recognition

Goal recognition is the task of identifying which goal (from a set of assumed possible ones) explain an observed **actor**'s behavior from some initial state of a given task environment (Sukthankar et al., 2014); it is considered a sub-problem of plan recognition (Pattison & Long, 2011). A common assumption of goal recognition is that the **observer** only looks at the actor's (unfolding) plan. An example goal recognition task within the `easy-grid` domain is illustrated in Figure 1b. Our formulation follows that by Ramírez & Geffner (2009), which casts the recognition task as a special-purpose classical planning one: goal recognition is tantamount to computing the plan that accounts for the input observation sequence at no extra cost to the actor's optimal plan to an assumed goal from their initial state (Masters & Sardina, 2019).

Thus, a goal recognition problem is a tuple $R = \langle \mathcal{L}, I, A, \mathcal{G}, \gamma, f_{\text{cost}}, \text{obs} \rangle$, where $\mathcal{L}$, I, A, γ, and f_{cost} are as before, $\mathcal{G}$ is a *set* of goals the actor is *assumed* to care to achieve satisfying $\forall G_i \in \mathcal{G}, G_i \subseteq \mathcal{L}$, and obs is the actor's *observed* plan (that is unfolding) satisfying $\forall a_i \in \text{obs}, a_i \in A$. The solution to this kind of goal recognition problem R is the (so-called) optimal goal set $\mathcal{G}^*$ satisfying $\forall G_i \in \mathcal{G}^*$: (a) $G_i \in \mathcal{G}$, (b) there is an optimal plan π_i^*, s.t. the final state $s_f \in trace(\pi_i^*, I)$ satisfies G_i; *i.e.* $G_i \subseteq s_f$, and (c) the plan π_i^* **observationally satisfies** obs. A plan $\pi = [a_1, ..., a_i, ..., a_m]$ observationally satisfies an action sequence obs $= [o_1, ..., o_j, ..., o_n]$, if there exists a strictly monotonic function $\mathcal{F}$ that maps the observation indices $j = 0, ..., n$ into action indices $i = 0, ..., m$, such that $a_{i=\mathcal{F}(j)} = o_j$; monotonicity guarantees π will contain the actions in obs *in the order they were observed*. Informally, a goal is deemed recognized if we can find a plan that can account for the observations at a cost equal to the cost of the optimal plan for the given goal. Formally, a solution to R is found via a **compilation-planning** procedure (Ramírez & Geffner, 2009), which we do not detail here.

The smaller the size of the optimal goal set $|\mathcal{G}^*|$, the better the solution is (i.e. the greater the solution's recall is). A key factor that contributes to the difficulty of a particular goal recognition problem – i.e. the difficulty in minimizing $\mathcal{G}^*$ – is *ambiguous behavior*: activity that plausibly contributes to the actor's pursuit of more than one goal. For example, in Figure 1b, the illustrated actor is pursuing the *most ambiguous plan*: it is not until the actor is at location `C6` that the actor *must* act in a way that betrays its intended goal; i.e. that collapses the size of $\mathcal{G}^*$ from 2 to 1. This is because the optimal plans toward achieving $g1$ and $g2$ individually share the prefix $\pi_{\text{most-ambiguous}} =$ [`(move Actor C1 C2)`, `(move Actor C2 C3)`, `(move Actor C3 C4)`, `(move Actor C4 C5)`, `(move Actor C5 C6)`]. In this case, the task environment's **worst-case distinctiveness** (Keren et al., 2019) is 5; i.e. the length of the *longest* **non-distinctive** (most ambiguous) **plan** is 5.

2.3 Goal Recognition Design

Goal recognition design (GRD) is precisely the problem of minimizing a goal recognition problem's worst-case distinctiveness (wcd) via modifications to the problem's **domain model**, defined with respect to a problem R as the tuple $R_{\text{M}} = \langle \mathcal{L}, I, A, \mathcal{G}, \gamma, f_{\text{cost}} \rangle$. Importantly, in the formulation of the goal recognition design problem, the modifications to the model are done *offline*. That is, a goal recognition domain model R_{M} is systematically modified to a new model R'_{M}, with minimum possible wcd (which is subsequently used to solve goal recognition problems).

Conceptually, the transformation seeks to minimize the longest possible non-distinctive plan, itself defined formally with respect to the domain model. A plan π is non-distinctive if it is a *prefix* to at least 2 plans, each one to a *distinct* goal in $\mathcal{G} \in R_{\text{M}}$. Formally, (similar to the definition of the optimal goal set $\mathcal{G}^*$) let $\Pi^*(G_i)$ be the set of optimal plans for goal $G_i \in \mathcal{G} \in R_{\text{M}}$. Then, a plan π is non-distinctive in R_{M} iff: (a) $\exists G_i, G_j \in \mathcal{G} \colon G_i \neq G_j$ (there are two distinct assumed goals), and (b) $\exists \pi_i \in \Pi^*(G_i), \pi_j \in \Pi^*(G_j) \colon \pi$ is a prefix of π_i and π_j (there is an optimal plan to each distinct goal and π is a prefix to both of them).

Of course, there may be multiple non-distinctive plans for a given goal recognition domain model. The non-distinctive plan we care about in GRD is the one that gives us the wcd: the longest (greatest cost) non-distinctive plan. If we let Π_{nd} denote the set of all non-distinctive plans, then,

$$wcd(R_{\text{M}}) = \max_{\pi \in \Pi_{\text{nd}}} f_{\text{cost}}(\pi) \tag{1}$$

A GRD problem is defined by specifying an *initial* goal recognition model R^0_{M} whose wcd should be minimized and a (so-called) **design model**: an encoding of the legal modifications to a given recognition problem R_i that result in a new problem R_j. The modifications *should* reduce the wcd of the input model. In other words, ideally, for all modifications that transform a model R^i_{M} into a new model R^j_{M}, $wcd(R^j_{\text{M}}) < wcd(R^i_{\text{M}})$. While Keren et al. (2019) propose four types of modifications, here we focus on one type – *action-removal* – deferring others to future work. Action-removal transformations center on modifying the set $A \in R_{\text{M}}$ to a set $A' \subset A$.

Definition 1 (Action-Removal Transformation) A tuple $\alpha = \langle R_{\text{M}}, a, R'_{\text{M}}, \Delta_{wcd} \rangle$, where R_{M} is a goal recognition model, a is the action to remove from $A \in R_{\text{M}}$, R'_{M} is the goal recognition problem that results by removing a from R_{M}, and Δ_{wcd} is the change in wcd that results by transforming R_{M} into R'_{M}. An action-removal transformation α is *proper* iff $\Delta_{wcd} > 0$.

We define a **classical GRD problem** as a GRD problem whose design model *only* contains action-removal transformations, per the initial formulation of GRD (Keren et al., 2014).

Definition 2 (Classical GRD Problem) A tuple $D = \left\langle R^0_{\text{M}}, \mathcal{D} \right\rangle$. R^0_{M} is the initial goal recognition domain model whose wcd should be minimized. $\mathcal{D}$ is a design model; *i.e.* a set of action-removal transformations as in Definition 1.

In practice, a classical GRD problem is not stated and solved explicitly. Instead, it is (also) solved via a compilation-planning procedure (Keren et al., 2019), which we do not detail here. The procedure effectively calculates the value Δ_{wcd} for all the actions in the domain, which is a key feature we rely on for goal elicitation planning: the technique we introduce next.

3. Goal Elicitation Planning: Eliciting Actor Goals via Embodied Observer Activity

Whereas goal recognition design asks "what domain model modifications must we introduce to minimize the wcd?", goal elicitation planning asks "what must an elicitor (née observer) agent do within the domain model to minimize the wcd?" Simply put, GEP is the *online* analogue of GRD.

We are motivated by goal reasoning within real-world contexts: while it is possible to design task environments that coerce actors to act in ways that reveal their intent (the motivation behind GRD), often the design of task environments is fixed *a priori*. For example, if we were attempting to predict the goals of a driver (or autonomous vehicle) who is navigating on the United States Interstate highway system, it is unreasonable to assume we will always be able to modify the highway circuit to elicit the goals of drivers. It is more reasonable to assume that we will be able to orchestrate the activity of other agents (e.g. law enforcement officers) to achieve states within the dynamics of the task environment that accomplish the same thing (e.g. setting up road-blocks).

Goal elicitation planning is conceptually simple and builds on planning, goal recognition, and (classical) goal recognition design formalisms: given a set of assumed target goals, minimize the task environment's wcd, by planning to achieve a state of the world that *effectively* performs an action-removal transformation. This is illustrated in Figure 1d: the elicitor in `E4` has planned to occupy `C4` (highlighted in blue), which will **clobber** (Boutilier & Brafman, 2001) the precondition `(isFree C4)` that is needed in order to execute the action `(move Actor C3 C4)` in the target's most ambiguous plan $\pi_{\text{most-ambiguous}}$ (illustrated in Figure 1b). This potentially reduces the wcd of the goal recognition problem from 5 to 2; in the scenario where the elicitor occupies `C4`, the target must abandon $\pi_{\text{most-ambiguous}}$ and from `C3` move to `B3` or `D3`, revealing their intended goal as being `(at A6)` or `(at E6)`, respectively (again, under the assumption of target optimality).

3.1 Goal Elicitation Plannability and Procedure

Goal elicitation plannability is a property of goal recognition *domain models* as defined in §2.3. We make the simplifying assumption that both the target and the elicitor are constrained to act in the same way. That is, both the target and the elicitor may act as given by the same set $A \in R_{\text{M}}$.

For a goal recognition domain model R_{M} to be **goal elicitation plannable** (GEPable), it must contain at *least* one action that can be **threatened** (Weld, 1994) by another. An action $a_{\text{threat}} \in A$ threatens another action $a_{\text{target}} \in A$ just when a_{threat} has an effect $\neg p \in \text{EFF}(a_{\text{threat}})$ whose dual p appears in the preconditions of a_{target}, *i.e.* $p \in \text{PRE}(a_{\text{target}})$; the action a_{target} is said to be *threatenable*. We denote the set of all such actions in a model as $A_{\text{threatenable}} \subseteq A \in R_{\text{M}}$.

To anticipate: in GEP, the target's most ambiguous plan should contain a_{target}, which the elicitor will aim to effectively remove by finding a plan whose *last* action is a_{threat}.

For convenience, we define a recognition model R_{M}'s **GEP number** as the number of threatenable actions: $|A_{\text{threatenable}}|$. Stated crisply, R_{M} is GEPable iff $|A_{\text{threatenable}}| > 0$. The GEP number informs a human operator whether GEP has the potential to be an effective method for reducing WCD. In the `easy-grid` task environment, the `move` operator from Listing 1 defines the template for the 98 actions available. Since every action instance has a ground version of the precondition literal `(isFree ?to)`, all actions are threatenable: they are threatened by any other action that has a matching `(not (isFree ?to))` in the effects. Thus, `easy-grid` is GEPable.

A threatenable action a_{target}'s target precondition $p \in \text{PRE}(a_{\text{target}})$ is what conceptually defines a GEP Problem in the context of a model R_{M}. Thus, a goal recognition domain model implicitly codifies several goal elicitation problems, one for each clobberable precondition of every threatenable action within R_{M}. A single GEP problem is defined by one such precondition, which we call the **elicitation condition**.

Definition 3 (GEP Problem) A tuple $E = \langle R_{\text{M}}, p \rangle$, where $R_{\text{M}} = \langle \mathcal{L}, I, A, \mathcal{G}, \gamma, f_{\text{cost}} \rangle$ is a goal recognition domain model, and $p \in \mathcal{L}$ is the *elicitation condition* satisfying $\exists a_{\text{target}}, a_{\text{threat}} \in A \colon p \in \text{PRE}(a_{\text{target}}) \wedge \neg p \in \text{EFF}(a_{\text{threat}})$.

While our `easy-grid` environment contains many GEP Problems, the one illustrated in Figure 1d is $E_{\text{1d}} = \langle$`easy-grid`, `(isFree C4)`$\rangle$. As illustrated, the elicitor in `E4` finds a plan – visualized by the blue highlighted path – to clobber `(isFree C4)` by occupying `C4` in advance of the target arriving. The elicitor's plan *effectively* performs a GRD-like action-removal. The removal is of the form $\alpha = \langle$`easy-grid`, `(move Actor C3 C4)`, `easy-grid`$', -3\rangle$.[1]

The elicitor's goal elicitation procedure – written precisely in Procedure 1 – searches for all the ways in which the elicitor can clobber the target's most ambiguous plan afforded by the goal recognition model. Thus, the elicitor does not search for a *single* plan, but rather for a *set* of plans. As we discuss later, an elicitor may need to be strategic about *how* it clobbers the target's plan.

The procedure first checks the model R_{M}'s GEP number to determine if it is GEPable (line 4). Importantly, we make no claims as to the relationship between the GEP number and the overall performance of the GEP procedure. We simply note that $|A_{\text{threatenable}}| > 0$ for GEP to be possible and envision future work of a problem formulation methodology that establishes a relationship between GEP performance and problem structure. If GEPable, the procedure proceeds to check the wcd reduction[2] obtained via the removal of each action in the model (lines 5-6). The procedure then creates several GEP problems for each action a_i whose removal is *proper* – i.e. whose $\Delta^i_{wcd} > 0$: one problem is created for each of a_i's preconditions (lines 7-8). Each GEP problem encapsulates a corresponding planning problem: the problem the elicitor must solve to clobber the target's most ambiguous plan. The procedure then finds a plan that solves that problem (line 9). If a plan is found, the procedure checks the last constraint the plan must satisfy (line 10): the clobbering plan must be *cheaper* than the most ambiguous plan *that the target will have executed up until that point*. For this, we need to be more precise about the task environment's dynamics and how the elicitor goes about choosing and executing their plan.

3.1.1 The Success of Goal Eliciting Plans

Whether or not the elicitor's goal eliciting plan is successful depends on the task environment's dynamics. From the perspective of the elicitor, the environment has an internal dynamic beyond its direct control that evolves per the target's activity. Thus, if the elicitor and the target *can* act in a state, what is the state that results when they *both* act?

1. In fact, occupying `C4` performs *multiple* action-removals within the `easy-grid` environment: namely, every ground instance of `(move ?actor ?from C4)` is removed.
2. While left implicit in the procedure, this involves using the GRD formulation to identify the potential reduction.

Procedure 1 A goal elicitation planning procedure, which allows an elicitor agent to actively formulate the goal of a target agent by acting to coerce the target to betray its goal.

Given: a goal recognition model $R_{\text{M}} = \langle \mathcal{L}, I, A, \mathcal{G}, \gamma, f_{\text{cost}} \rangle$.
Find: a set of clobbering plans Π satisfying $\forall \pi_p \in \Pi\colon f_{\text{cost}}(\pi_p) \leq f_{\text{cost}}(\textit{plan-until}(\pi_{wcd}, I, p))$.

procedure GOAL-ELICITATION-PLAN-SEARCH(R_{M})
  Let $\pi_{wcd} \leftarrow$ the plan whose length is $wcd(R_{\text{M}})$ ▷ The most ambiguous plan afforded by R_{M}.
  Let $\Pi \leftarrow \varnothing$ ▷ Start with an initially empty solution set.
  if $|A_{\text{threatenable}}| > 0$ **then** ▷ If the model R_{M} is goal elicitation plannable,
    for each action $a_i \in A \in R_{\text{M}}$ **do** ▷ then for each action a_i in the model,
      if $\Delta^i_{wcd} > 0$ **then** ▷ check if removing a_i reduces the wcd (the action-removal is *proper*).
        for each precondition $p_j \in \text{PRE}(a_i)$ **do** ▷ If so, then for each elicitation condition in a_i,
          Let $E_{ij} \leftarrow \langle R_{\text{M}}, p_j \rangle$ ▷ generate a GEP problem for the elicitation condition p_j.
          Let $\pi_{ij} \leftarrow$ a solution to the planning problem $\langle \mathcal{L}, I, A, \{\neg p_j\}, \gamma, f_{\text{cost}} \rangle$ ▷ Try clobbering p_j.
          if $f_{\text{cost}}(\pi_{ij}) \leq f_{\text{cost}}(\textit{plan-until}(\pi_{wcd}, I, p_j))$ **then** ▷ If it can clobber p_j before it is needed,
            $\Pi \leftarrow \Pi \cup \pi_{ij}$ ▷ then the clobbering plan π_{ij} is a potential solution.
  return Π

In this work, we assume a game-theoretic planning multi-agent model (e.g. Brafman et al., 2009). This model fully partitions the states of the task environment into two sets: **elicitor states** and **target states**. In the former, only the elicitor may act and in the latter only the target may act. In game-theoretic terms, the task environment is *symmetric* (agents have the same moves available), and *sequential* (agents take turns).

For an elicitor's plan to succeed – and thus for a such a plan to be a solution to a GEP problem – their clobbering plan must be *cheaper* than the most ambiguous plan *that the target will have executed up until that point*. Given the turn-taking model, we now precisely state what that means. For convenience, we define a function $\textit{plan-until}(\pi, I, p)$, which denotes a subsequence π_{sub} of π – thus, $\textit{plan-until}(\pi, I, G) = \pi_{\text{sub}} \subseteq \pi$ is itself a plan – for which the last state s_m in $trace(\pi_{\text{sub}}, I)$ is the state in which p is *used* as a precondition for the remaining plan ($\pi \setminus \pi_{\text{sub}}$).

We assume the target goes *first*. Let π_{wcd} denote the target's longest non-distinctive plan in R_{M} and suppose we have an elicitor's plan π_p that clobbers p for the GEP problem $E = \langle R_{\text{M}}, p \rangle$. In essence, π_p is a GEP solution if its last step executes prior to when p is first needed in the π_{wcd} plan.

Definition 4 (GEP Solution) A plan $\pi_p = [a_1, ..., a_{\text{threat}}]$ is a solution to a GEP problem $E = \langle R_{\text{M}}, p \rangle$ with model $R_{\text{M}} = \langle \mathcal{L}, I, A, \mathcal{G}, \gamma, f_{\text{cost}} \rangle$ just when: (a) $\neg p \in \text{EFF}(a_{\text{threat}})$, and (b) $f_{\text{cost}}(\pi_p) \leq f_{\text{cost}}(\textit{plan-until}(\pi_{wcd}, I, p))$.

The function $\textit{plan-until}(\pi_{wcd}, I, p)$ denotes a subsequence $\pi_{\text{sub-wcd}} \subseteq \pi_{wcd}$ for which the last state in $trace(\pi_{\text{sub-wcd}}, I)$ is the first one in which p is subsequently needed. In other words, it returns the target's plan up until p is *first* needed by the target's remaining plan. Thus, if the elicitor's plan π_p is cheaper to execute than that subsequence, the elicitor will force the target to abandon the remaining (unexecuted) portion of π_{wcd}, making π_p a solution.

In our example, the target's plan π_{wcd} (which starts executing first) first needs `(isFree C4)` after the first two actions `(move Actor C1 C2)` and `(move Actor C2 C3)`; the subsequent action `(move`

`Actor C3 C4)` has `(isFree C4)` as a precondition. Thus, $plan\text{-}until(\pi_{wcd}, I, \texttt{(isFree C4)}) = \pi_{\text{sub-wcd}} = [\texttt{(move Actor C1 C2)}, \texttt{(move Actor C2 C3)}]$. Further, $f_{\text{cost}}(\pi_{\text{sub-wcd}}) = 2$.

The elicitor's plan in Figure 1d is `[(move Elicitor E4 D4), (move Elicitor D4 C4)]`; this plan accomplishes `(not (isFree C4))` via the last action $a_{\text{threat}} =$`(move Elicitor D4 C4)` (first GEP solution condition), and has a cost $f_{\text{cost}}(\pi_{\texttt{(isFree C4)}}) = 2 \leq f_{\text{cost}}(\pi_{\text{sub-wcd}})$ (second GEP solution condition). Thus $\pi_{\texttt{(isFree C4)}}$ is a solution plan to the GEP problem $E_{1\text{d}} = \langle \texttt{easy-grid}, \texttt{(isFree C4)} \rangle$.

3.1.2 The Choice of Which Goal Eliciting Plan to Execute

Procedure 1 determines a set of plans that the elicitor may carry out to elicit the goals of a target. We assume that the elicitor is acting *lazily*: that is, the elicitor picks a plan that works "just-in-time" to clobber the target's most ambiguous plan. Thus, the elicitor's plan π_p will align with the target's plan such that a_{threat} executes immediately before condition p is needed. We therefore assume that whenever the elicitor *may* act, the elicitor may opt to *not* act; for this, the task environment must support some dummy `(no-op ?agent)` action with no preconditions or effects that simply consumes the elicitor's turn to enable such an alignment. Equivalently, the elicitor may act *eagerly* and execute the plan up until π_p is clobbered and *then* perform a `(no-op ?agent)` action until the target's plan is derailed. This assumption, coupled with the turn-taking assumption, lets us treat GEP solutions as ones of **achievement** (Vattam et al., 2013): achieving goals with no temporal duration, satisfied by the instantaneous satisfaction of some goal condition. Future work might expand our model to consider **maintenance goals**, where the GEP problem's clobbering condition ought be maintained for some temporal extent (possibly forever).

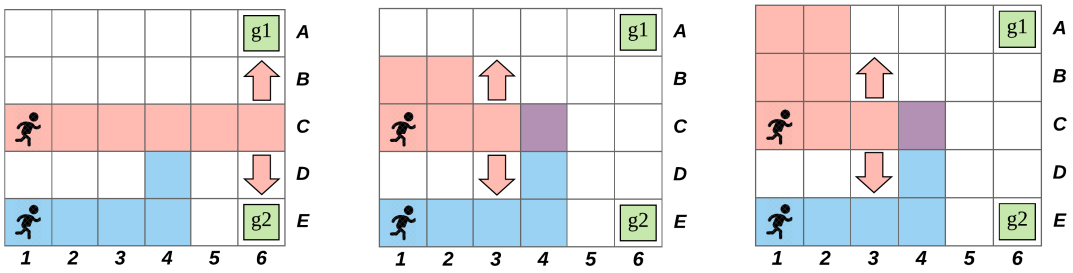
There is some flexibility between the extremes of acting *lazily* and acting *eagerly*. To anticipate our later discussion: if there is *any* need for a `(no-op)` action, the elicitor's choice of which plan to execute becomes *strategic*. The elicitor may choose to act cost-sub-optimally for as many moves as there are `(no-op)`'s, and what the elicitor does could be guided by (for instance), the elicitor's anticipation of the target's observation of the elicitor. We return to this theory-of-mind reasoning – which we posit may have material consequence for the elicitor – after we discuss what it means for conducting GEP in non-cost optimal contexts.

3.2 From Cost-Optimal to Bounded Non-Optimal Agents

To account for non-cost-optimal (i.e. non-optimal) agents, we follow the strategy by Keren et al. (2015), and define non-optimality as a *bounded rationality* measure. Thus, a non-optimal target agent has a budget, $b \in \mathbb{R}^{>0}$, enabling them to pursue non-optimal plans up to $f_{\text{cost}}(\pi^*) + b$ to their goals. The budget b is added as a dimension to the GEP problem, yielding a **bounded non-optimal GEP problem**.

Definition 5 (Bounded Non-optimal GEP Problem) A tuple $E_b = \langle R_{\text{M}}, p, b \rangle$, where $R_{\text{M}} = \langle \mathcal{L}, I, A, \mathcal{G}, \gamma, f_{\text{cost}} \rangle$ is a goal recognition domain model, $p \in \mathcal{L}$ is the *elicitation condition* as in Definition 3, and $b \in \mathbb{R}^{>0}$ is the target's non-optimal behavior *budget*.

GRD accounts for non-optimal agents when calculating wcd by showing that paths only increase in cost from the optimal plan and therefore the same wcd reductions apply. For GEP, the increase



(a) The elicitor *cannot* find a plan π_p to threaten the target's plan π_{wcd}.

(b) The elicitor *can* find π_p for a non-optimal target with $b = 2$.

(c) The elicitor has greater flexibility for a non-optimal target with $b = 4$.

Figure 2: An elicitor is able to achieve increasingly flexible GEP solutions as the non-optimal budget of the target increases. In (b) and (c), the elicitor targets the removal of action `(move Agent C3 C4)` to reduce the wcd, whereas the target goes as far toward row A as the budget allows.

in plan cost affords an elicitor more leeway in GEP solution cost. The elicitor can therefore achieve a wcd reduction for more actions. We account for the greater number of wcd reductions due to non-optimality by defining more GEP solution types.

Figure 2 illustrates the effect of permitting the target to exhibit bounded non-optimal behavior in our original `easy-grid` environment. Here, the elicitor starts in a new location `E1`. In Figure 2a, the budget $b = 0$, making it impossible to arrive at `C4` in time to force the target to reveal their goal. In this case, our GEP agent would fail to achieve the same wcd reduction as GRD.

When the target's budget is 2, however, they are *allowed* to deviate from the optimal path (Figure 2b). In turn, the elicitor can now potentially solve the GEP problem: a wcd reduction by removing `(move Agent C3 C4)` is *minimally possible* if the target uses their entire budget. The corresponding clobbering plan is considered a **minimal GEP solution**, because the elicitor will have *just enough* time to clobber `(isFree C4)`. For clarity, let $\pi_{wcd}^{\text{until-}p} = \mathit{plan\text{-}until}(\pi_{wcd}, I, p)$. Then:

Definition 6 (Minimal GEP Solution) A plan $\lfloor \pi_p^b \rfloor = [a_1, ..., a_{\text{threat}}]$ is a minimal solution to a bounded non-optimal GEP problem $E_b = \langle R_{\text{M}}, p, b \rangle$ with model $R_{\text{M}} = \langle \mathcal{L}, I, A, \mathcal{G}, \gamma, f_{\text{cost}} \rangle$ just when: (a) $\neg p \in \text{EFF}(a_{\text{threat}})$, and (b) $f_{\text{cost}}(\lfloor \pi_p^b \rfloor) = f_{\text{cost}}(\pi_{wcd}^{\text{until-}p}) + b$.

If we increase the target's budget to 4 (Figure 2c), the elicitor is afforded even more opportunity to prevent movement. If the target uses their entire budget, the elicitor will arrive at `C4` with several moves to spare. The elicitor's clobbering plan is considered a **partial GEP solution**; in effect it provides a "(cost) window of opportunity" for the elicitor, which may or may not materialize given the target's actions. The cost window's lower bound is $f_{\text{cost}}(\pi_{wcd}^{\text{until-}p})$, and its upper bound is $f_{\text{cost}}(\pi_{wcd}^{\text{until-}p}) + b$.

Definition 7 (Partial GEP Solution) A plan $[\pi_p^b] = [a_1, ..., a_{\text{threat}}]$ is a partial solution to a bounded non-optimal GEP problem $E_b = \langle R_{\text{M}}, p, b \rangle$ with model $R_{\text{M}} = \langle \mathcal{L}, I, A, \mathcal{G}, \gamma, f_{\text{cost}} \rangle$ just when: (a) $\neg p \in \text{EFF}(a_{\text{threat}})$, and (b) $f_{\text{cost}}(\pi_{wcd}^{\text{until-}p}) < f_{\text{cost}}([\pi_p^b]) < f_{\text{cost}}(\pi_{wcd}^{\text{until-}p}) + b$.

Table 1: The GEP Number for each problem in the `easy-grid` and modified `easy-grid′` domains.

Domain	**GEP Number**				
	p01 (*Problem-01*)	p02	p03	p04	p05
`easy-grid`	0	0	0	0	0
`easy-grid′`	160	76	108	102	102

4. Evaluation

Our evaluation has two aims. The first aim is to show that domains have inherent properties that make them conducive to GEP as we described in Section 3.1. The modifications we made to the benchmark `easy-grid` to increase its GEP number are summarized in Table 1; these modifications result in the actual domain used for evaluation, which we refer to as `easy-grid′`. Our second aim is to demonstrate that achieving a minimum wcd at run time is possible using the GEP constructs presented in Section 3. We provide preliminary results for wcd reduction using optimal and bounded non-optimal agents, using the `easy-grid′` domain and problems.

Domains We use a modified benchmark domain from those used by Ramírez & Geffner (2009) for goal recognition as classical planning. Specifically, we use the five problems from the `easy-grid` domain; while simple, the domain supports our evaluation goals. Recall that for GEP to be possible, the problem's GEP number must be greater than 1. We therefore modified the original `easy-grid` domain by modifying its `move` operator in two ways; the net result is the operator that appears in Listing 1. First, we added an `?agent` parameter to afford multiple distinct agents to take the same action. Second, we added the predicate `(isFree ?loc)` to its preconditions as well as `(not (isFree ?loc))` to the effects. These changes ensure that two agents cannot co-occupy the same place and increases the GEP number as indicated in Table 1.

Procedure and Results We implemented Procedure 1 and applied it to the `easy-grid′` benchmark problems. We focus on action removals as it is the foundation from which we can build GEP versions of the remaining three GRD domain modifications (action conditioning, single-action sensor refinement and sensor placement) Our software uses the public software repositories for both GRD and the Fast Downward (FD) planner (Helmert, 2006). We use the GRD software to calculate the wcd for each problem. Our implementation is publicly available at https://anonymous.4open.science/r/c8e1b370-9657-4001-8cac-a734230f6581/.

Optimal Agents Figure 3 contains the minimum wcd when applying GEP for optimal agents in the `easy-grid′` domain. We compare GEP results to the initial wcd (with no GEP or GRD) and the minimum wcd achieved using GRD action removals. In problems p02 and p03, neither GEP or GRD improves the initial wcd. This lack of wcd improvement is a function of the problem design where goal achievement pathways are very ambiguous and do not include any actions that improve goal elicitation. Conversely, in p01 and p05 both GEP and GRD improve the minimum wcd by 1. While only a small wcd improvement, it does demonstrate that GEP can achieve online performance equivalent to the offline GRD. Lastly, in p04 we observe a case when GEP does not achieve the same

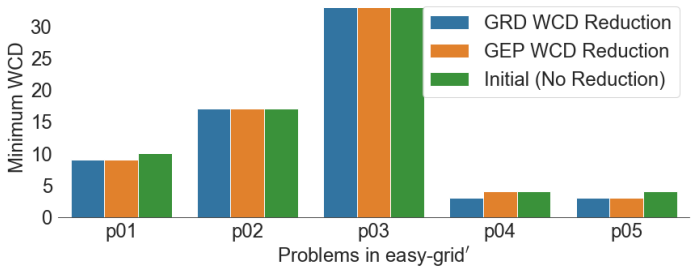


Figure 3: wcd for `easy-grid`′ obtainable via GEP (online, via an elicitor agent), GRD (offline), and baseline (no reduction), for an optimal target agent.

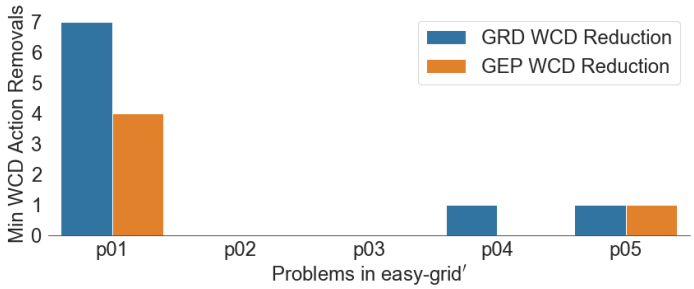


Figure 4: Number of actions that achieve the minimum wcd using GRD/GEP for optimal targets.

minimum wcd as GRD. This discrepancy emerges because the elicitor cannot generate a plan (via FD) with a cost less than 4. Thus, the elicitor arrives "too late" to block the target and is not able to reveal their goal as early as GRD does.

Figure 4 contains the number of actions that achieve the minimum wcd in each problem. In p01, GRD and GEP both achieve the minimum wcd but GRD has more actions that do so (10 v. 7, respectively). This is because the elicitor is unable to find plans with a cost less than 9 for some of the actions. In problems p02 and p03, neither GRD nor GEP achieved the minimum wcd and therefore no actions achieve minimum wcd. In p04, GRD achieves the min wcd but only with a single action compared to GEP that does not achieve the min wcd at all. Lastly, in p05 both methods achieve the min wcd with a single action.

Bounded Non-Optimal Agents As before, bounded non-optimal agents have a budget allowing them to deviate from optimal plans toward their goals. Figure 6 contains the results of applying GEP to the `easy-grid`′ domain for non-optimal agents with a budget of 2. Here, GEP now achieves the same wcd as GRD in p04. This improvement is due to the non-optimality of the target, which increases the window for a (minimum) GEP solution to threaten an action and obtain a maximal wcd reduction. Results for p01 and p05 remain equivalent for GEP and GRD. Like the optimal target agent results, neither GEP nor GRD improves upon the minimum wcd in p02 and p03.

Like before, we investigated the number of actions that achieve the minimum wcd. Figure 5 contains the number of actions that achieve the minimum wcd in each problem for non-optimal

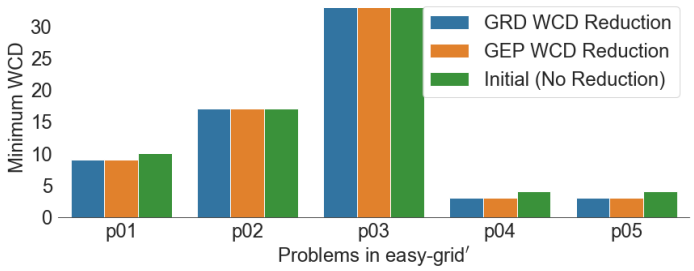


Figure 5: wcd for `easy-grid`′ obtainable via GEP (online, via an elicitor agent), GRD (offline), and baseline (no reduction), for a bounded non-optimal target agent with (budget) $b = 2$.

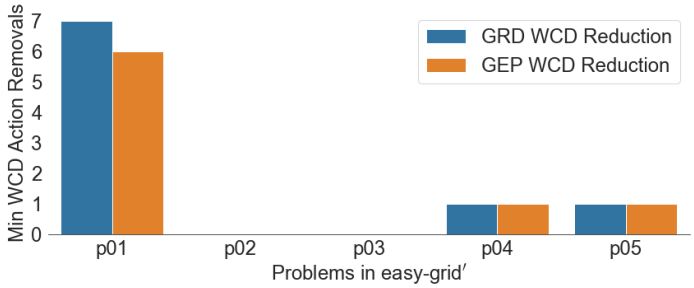


Figure 6: Number of actions that achieve the minimum wcd using GRD/GEP for bounded non-optimal targets with (budget) $b = 2$.

agents. We note that in p01, GRD and GEP both achieve the minimum wcd but GRD has more actions that do so (10 v. 9, respectively). This is due to the elicitor being unable to find plans with a cost less than 9 for some of the actions. This has increased in comparison to the optimal agent, as the bounded non-optimal agent provides the GEP agent an extra plan cost buffer needed to threaten a move action. In problems p02 and p03, neither GRD nor GEP achieved the minimum wcd and therefore there are no actions. In p04 and p05, both GRD and GEP achieve the min wcd with a single action. Again, due the non-optimal agent, the elicitor can now arrive in time to achieve the min wcd with a single action whereas they did not in the optimal case.

Discussion Our results provide evidence that GEP *can* achieve minimum wcd reductions equivalent to GRD for both optimal and bound non-optimal target agents in the `easy-grid`′ benchmark problems. In p04, we identified a case where the target agent type (i.e. optimal v. bounded non-optimal) affects whether GEP achieves the minimum wcd reduction. Further, in p01 we show that while GEP achieves the same wcd reduction, it has fewer options (actions) to choose from. This is due to plan cost being greater than the minimum wcd and the elicitor being unable to arrive "in time" to prevent the key action from being executed. Once again, the target agent type affects the number of actions a GEP agent can threaten. A non-optimal agent provides a greater window of opportunity for the elicitor to threaten key actions.

We also found that domains have an inherent property, their GEP number, that makes them amenable to favorable GEP results, such as `easy-grid`. GEP is more effective in domains with actions whose preconditions are threatened by other actions. Negating preconditions of other actions gives an elicitor greater coverage of removed actions and more options to generate a GEP solution. While we made a reasonable modification to the `easy-grid` domain to demonstrate these results, we expect to formalize this handcrafted approach with an automated problem formulation method in future work.

5. Related Work

GEP is a technique that allows an agent to actively formulate the goals of another. Extant research within the goal reasoning community has primarily focused on *self*-goal formulation (e.g. Wilson et al., 2013) or on the *passive* formulation of another's goals (discussed below).

As discussed, an agent's formulation of the goals of another is one of the central concerns of the PAIR community. However, *most* methods assume that the observer is external to the environment being analyzed. Thus, GEP differs from extant research in PAIR in a significant way: in a PAIR-sense, GEP enables an observer to *proactively* take actions in their environment to recognize the actor's goal, as opposed to passively observing their actions.

Two nascent bodies of work have departed from the trend of disembodied PAIR, both (confusingly) referred to by the same name: **active goal recognition**. The first, by Amato & Baisero (2019), interleaves goal recognition and planning: an embodied PAIR agent must balance completion of their own tasks with information gathering about the behavior of some other agent. In GEP, however, the PAIR agent (elicitor) is planning for the express purpose of information gathering about the behavior of some other agent (the target). The second, by Shvo & McIlraith (2019) is closer to our work but still distinct. Their PAIR agent can perform two kinds of sensing actions: those that sense actions (the standard approach in PAIR) and those that sense actions (a novel kind of sensing). By sensing states and relying on task environment **landmarks** – states that must appear in every solution of a planning problem – their PAIR agent can better eliminate and confirm goal-hypotheses. However, they defer actively impeding or aiding the target to future work, which is precisely the work we do here.

The goal reasoning community uses PAIR methods to inform action. Pozanco's 2018 work on counterplanning –passively identifying goals then actively blocking them– is related in that elicits goals but in a passive manner using landmarks. GEP differs by actively eliciting goals without using landmarks and is currently socially agnostic, neither blocking or aiding the target, as opposed to blocking the target's goals as in counterplanning. Active behavior recognition (Alford et al., 2018) reduces the ambiguity of another agent's goal by taking actions. Actions are determined through a case-based reasoning method and reduce the ambiguity of a target's goal in a probabilistic manner. Active behavior recognition differs from GEP in its probabilistic assessments of target goals and reliance on historical behaviors in a case library.

Finally, GEP generalizes the (turn-taking) adversarial game-theoretic planning model (*e.g.* Brafman et al., 2009), and at the same time constrains it: in GEP, the game-theoretic utility of the elicitor is *always* couched with respect to successfully eliciting the target's goal. While we have not

explored a reified notion of utility in this paper, we note that a non-cost-optimal elicitor agent must carefully balance the utilities of acting or not toward clobbering a GEP problem's elicitation condition. While eliciting the target's goal is (arguably) of highest utility to an elicitor, the elicitor may want to account for the target observing the elicitor themself; the elicitor may want to avoid acting in a manner that betrays *which* clobbering condition they are pursuing, or may want to act to suggest the *wrong* clobbering condition in the mind of the target thereby "throwing the target off."

6. Conclusion

Discovery within and adapting to dynamic environments is a key feature of goal reasoning agents. As such, recognizing another agent's goals is an essential social function necessary for coordinating and managing goals. While passively observing and inferring a target agent's goal has been the modus operandus of PAIR methods, a goal reasoning agent has the ability to act within their environment, shaping it to reveal another agent's goal and manage it with other priorities. We have leveraged a goal-reasoning agent's ability to act in their environment along with GRD's PAIR framework to create a new capability called GEP. Goal reasoning agents now have the the capability to take actions, instead of only observing, that reveal a target agent's goals at the earliest possible time.

Our GEP formalization extends the existing GRD framework by making wcd reductions online and includes three contributions. First, we show that GEP is domain-problem independent. A problem's GEP number represents the number of actions that can be threatened. This is an essential property of a problem as it provides a GEP-enabled agent the options to prevent actions being executed by the target agent, coercing the target to reveal their goals. For a problem to be GEP-able, its GEP number needs to be greater than zero. Second, GEP achieves similar performance to its offline counterpart, GRD. We show that for GEP to achieve a wcd reduction for a given action, GEP must threaten the action with a GEP solution plan with a plan cost less than the wcd that would result from removing the action entirely. Results from the `easy-grid`$'$ benchmark problems show that GEP achieves an online wcd reduction equivalent to it's offline counterpart, GRD, in four of the five problems when the target agent behaves cost-optimally. Our third contribution shows that GEP extends beyond identifying cost-optimal agent goals. Non-optimal agents have a cost budget which they can use to deviate from the optimal path towards their goals. This extra window of time, when used by the target agent, affords GEP the opportunity to execute plans that did not successfully threaten the action in time under the optimal agent assumption. We define partial-achievement and minimal-achievement GEP solution plans to refine under what assumptions GEP can achieve different wcd reductions. Our `easy-grid`$'$ benchmark results show that when the non-optimal agent uses a budget of 2, GEP achieves the same wcd reductions as GRD across the benchmark problems.

There are several avenues for immediate investigation. Foremost is to integrate the capability into existing goal reasoning agent frameworks. Both the ActorSim framework (Roberts, 2016) and the MIDCA meta-cognitive architecture (Cox et al., 2016) are intriguing candidates. Another key investigation is to automate a problem formulation methodology that contains the GEP domain performance criteria. Automating the criteria will enable GEP performance estimates and provide insights into domain modifications so GEP can be applied to additional benchmarks and more practical applications such as games.

References

Alford, R., Borck, H., Karneeb, J., & Aha, D. W. (2018). Active Behavior Recognition in Beyond Visual Range Air Combat. *International Joint Conference on Artificial Intelligence Workshop on Goal Reasoning* (pp. 1–14).

Amato, C., & Baisero, A. (2019). Active goal recognition. *arXiv preprint arXiv:1909.11173*.

Amos-Binks, A., & Dannenhauer, D. (2020). Anticipatory Thinking: A metacognitive capability. *AAAI Fall Symposium on Cognitive Systems for Anticipatory Thinking - CEUR Workshop Proceedings*, *2558*.

Amos-Binks, A., Dannenhauer, D., & Aha, D. (2019). Intention Dynamics of Rebel Agent Behavior. *Advances in Cognitive Systems*, *7*.

Bonet, B., & Geffner, H. (2001). Planning as heuristic search. *Artificial Intelligence*, *129*, 5–33.

Boutilier, C., & Brafman, R. I. (2001). Partial-order planning with concurrent interacting actions. *Journal of Artificial Intelligence Research*, *14*, 105–136.

Brafman, R. I., Domshlak, C., Engel, Y., & Tennenholtz, M. (2009). Planning games. *Proceedings of the 21st International Joint Conference on Artificial Intelligence*.

Bratman, M. (1987). *Intention, Plans, and Practical Reason*. Palo Alto: Center for the Study of Language and Information.

Cohen, P. R., & Levesque, H. J. (1990). Intention is choice with commitment. *Artificial Intelligence*, *42*, 213–261.

Cox, M. T., Alavi, Z., Dannenhauer, D., Eyorokon, V., Munoz-Avila, H., & Perlis, D. (2016). Midca: A metacognitive, integrated dual-cycle architecture for self-regulated autonomy. *Proceedings of the 30th AAAI Conference on Artificial Intelligence*.

Fikes, R. E., & Nilsson, N. J. (1971). STRIPS: A new approach to the application of theorem proving to problem solving. *Artificial intelligence*, *2*, 189–208.

Grosz, B. J., & Kraus, S. (1999). The evolution of sharedplans. In *Foundations of rational agency*, 227–262. Springer.

Helmert, M. (2006). The fast downward planning system. *Journal of Artificial Intelligence Research*, *26*, 191–246.

Jokinen, K. (1996). Goal formulation based on communicative principles. *Proceedings of the 16th Conference on Computational Linguistics* (pp. 598–603). Association for Computational Linguistics.

Keren, S., Gal, A., & Karpas, E. (2014). Goal recognition design. *Proceedings of the 24th International Conference on Automated Planning and Scheduling*.

Keren, S., Gal, A., & Karpas, E. (2015). Goal recognition design for non-optimal agents. *Proceedings of the 29th AAAI Conference on Artificial Intelligence*.

Keren, S., Gal, A., & Karpas, E. (2019). Goal recognition design in deterministic environments. *Journal of Artificial Intelligence Research*, *65*, 209–269.

Masters, P., & Sardina, S. (2019). Goal recognition for rational and irrational agents. *Proceedings of the 18th International Conference on Autonomous Agents and Multi-Agent Systems* (pp. 440–448).

McDermott, D., Ghallab, M., Howe, A., Knoblock, C., Ram, A., Veloso, M., Weld, D., & Wilkins, D. E. (1998). *Pddl–the planning domain definition language*. Technical report, Yale Center for Computational Vision and Control, New Haven, CT, USA.

Muñoz-Avila, H., Dannenhauer, D., & Reifsnyder, N. (2019). Is everything going according to plan? expectations in goal reasoning agents. *Proceedings of the AAAI Conference on Artificial Intelligence* (pp.

9823–9829).

Pattison, D., & Long, D. (2011). Accurately determining intermediate and terminal plan states using bayesian goal recognition. *Proceedings of the 1st Workshop on Goal, Activity and Plan Recognition at the 21st International Conference on Automated Planning and Scheduling*.

Pozanco, A., Fernández, S., & Borrajo, D. (2018). Learning-driven goal generation. *AI Communications*, *31*, 137–150.

Ramírez, M., & Geffner, H. (2009). Plan recognition as planning. *Proceedings of the 21st International Joint Conference on Artificial Intelligence*.

Roberts, M. (2016). Goal Reasoning, Planning, and Acting with ACTORSIM, The Actor Simulator. *Advances in Cognitive Systems*, *21*, 273–280.

Shvo, M., & McIlraith, S. A. (2019). Active Goal Recognition. *AAAI Conference on Artificial Intelligence*. From http://arxiv.org/abs/1909.11173.

Sukthankar, G., Geib, C., Bui, H. H., Pynadath, D., & Goldman, R. P. (2014). *Plan, Activity, and Intent Recognition: Theory and Practice*. Elsevier.

Vattam, S., Klenk, M., Molineaux, M., & Aha, D. W. (2013). Breadth of approaches to goal reasoning: A research survey. *Proceedings of the Workshop on Goal Reasoning at the 2nd Annual Conference on Advances in Cognitive Systems*.

Weld, D. S. (1994). An introduction to least commitment planning. *AI Magazine*, *15*, 27–61.

Wilson, M. A., Molineaux, M., & Aha, D. W. (2013). Domain-independent heuristics for goal formulation. *Proceedings of the 26th International FLAIRS Conference*.