Strathprints Institutional Repository

Pattison, D. and Long, D. (2010) *Domain independent goal recognition.* [Proceedings Paper]

Domain Independent Goal Recognition

David Pattison and **Derek Long**[1]

Abstract. Goal recognition is generally considered to follow *plan* recognition. The plan recognition problem is typically defined to be that of identifying which plan in a given library of plans is being executed, given a sequence of observed actions. Once a plan has been identified, the goal of the plan can be assumed to follow. In this work, we address the problem of goal recognition directly, without assuming a plan library. Instead, we start with a domain description, just as is used for plan construction, and a sequence of action observations. The task, then, is to identify which possible goal state is the ultimate destination of the trajectory being observed.

We present a formalisation of the problem and motivate its interest, before describing some simplifying assumptions we have made to arrive at a first implementation of a goal recognition system, AUTOGRAPH. We discuss the techniques employed in AUTOGRAPH to arrive at a tractable approximation of the goal recognition problem and show results for the system we have implemented.

1 INTRODUCTION

Goal Recognition (GR) is the process of inferring an agent's end goals given a series of observed actions. This is clearly related to the *Plan Recognition* (PR) problem which aims to also find the plan being executed. *Planning* is simply the generation of these plans in an efficient and sensible manner. Yet despite both being based on *actions*, *states* and *goals*, and effectively mirroring one another, advances in research have rarely overlapped.

Previous work has often focused on a single application of the recognition problem, such as identification of human goals through observation of behaviour [15], giving speech/text context [9] or responding with natural dialogue [20]. These have all resulted in systems and algorithms that lack generality or widespread application.

AUTOGRAPH (AUTOmatic Goal Recognition with A Planning Heuristic), is a new approach to Goal Recognition which makes use of Planning techniques. The system uses a standard planning domain model, avoiding the construction of a goal/plan library.

2 MOTIVATION AND PRIOR APPROACHES

Plan and Goal Recognition problems are motivated by the desire to anticipate the actions or objectives of an agent that is being observed. There are many situations in which this could be useful, including detection and prevention of crime, in teaching, in monitoring the elderly or infirm in their own homes, in military operations and in games. In computer games, intelligent responses to human player activity depend on recognising what that activity might be. Creating a believable and responsive environment that allows players to participate in a truly immersive experience requires that computer controlled agents react to human players with plausible levels of understanding of the human players' actions. This context, in particular, motivates two assumptions underlying our work: first, that the actions are fully observable (game software mediates every action on behalf of the players) and, second, that we are interested in identifying goals as early as possible during the execution of the plan.

Kautz [17] defines the plan recognition problem as minimising the number of top-level, hierarchical plans which explain a sequence of observed actions. Plans were taken from a *plan graph* and every action is assumed to be *relevant* to the plan being executed. The library containing known, valid plans has remained a key element of plan recognition ever since. This structure presents several drawbacks such as the time, effort and space required to construct it and its inevitable incompleteness and irrelevant content. AUTOGRAPH attempts to address these problems in three areas: *Completeness*, *Scalability* and *Domain Independence*.

Completeness: It is impossible to generate and store every valid plan in a library for non-trivial problems. Previous work has often made use of tree-like structures to represent a large number of plans efficiently but cannot hold *all* possible plans or goals. In our work, any conjunction of literals may form a hypothesis.

Scalability: The scaling behaviour of plan recognition systems is highly dependent on library sizes. This and the previous problems combine to create a tension between scalability and completeness.

Domain Independence: Generating plan libraries is time-consuming and restricts application to domains for which libraries are available.

The goal recognition system of Blaylock and Allen [1] does allow unseen plans to be recognised but must first be *trained* using valid plans *and* explicitly defined goals. Lesh and Etzioi have also explored adapting recognition to a previously unseen plan with the *ADAPT* system [19, 18]. The recogniser is trained using recent behaviour which has **not** been annotated with the true goal. This data is then used to try and find the combination which provides the best results. Unlike the previous system, this work does not assume access to training data or require a policy to be constructed for recognition.

We are not the first to propose using *planning* for recognition. Hong [14] proposes an approach in which, as actions are observed, a *goal graph* is constructed similarly to a *plan graph*, with propositions which are recognised goals being linked into a goal layer. The system scales well, but must be provided with an explicit set of valid goals to be used in the graph construction and analysis processes.

Most recently Ramírez and Geffner [22] make use of *heuristic estimation* to eliminate goals from a candidate set, due to an increasing heuristic distance from the current state. The authors present two approaches, assuming *optimal* or *suboptimal* plans, with goals that have become impossible being removed from the *possible* set. Once candidate goals have been eliminated they are never reconsidered. Ramírez and Geffner assume a (small) set of possible goal states is supplied explicitly and they work with an assumption of *partial* observability, so that only a subsequence of the plan is observed.

[1] Department of Computer and Information Science, University of Strathclyde, Glasgow G1 1XH, UK, email: {david.pattison, derek.long}@cis.strath.ac.uk

3 PROBLEM DEFINITION

We now formally define the goal recognition problem. We start with the same framework that is used in classical planning, based on a propositional action model structure. A goal recognition problem is based on a standard planning problem (the facts, actions and initial state). Of course, the goal recognition problem does not contain a goal specification — the problem is to find this specification.

Definition 1. *Goal Recognition Problem Base*
A goal recognition problem base is a triple $\langle F, A, I\rangle$, where F is a set of primitive (propositional) facts, A is a set of actions and $I \subseteq F$ is the initial state for the problem. Each action $a \in A$ is a triple $\langle pre_a, add_a, del_a\rangle$, where $pre_a, add_a, del_a \subseteq F$ are the preconditions, add effects and delete effects of a, respectively.

In addition to the base, a goal recognition problem requires observations: a sequence of actions. We assume that all actions and states are fully observable, but we want to identify the goals as early as possible during execution of the plan. Before we define the goal recognition problem, however, we briefly consider the nature of the solutions we seek and the implications this has on the problem itself. Our expectation is that we should be presented with a goal recognition problem base and a series of actions, with the objective being to identify the target goals of the agent performing the actions. We assume that the agent actually has a target and is not simply executing actions at random. However, even though the agent has an objective, it is not clear that the actions we observe will be unambiguously leading the agent towards this. To simplify things, we begin by assuming that the agent is sufficiently intelligent to make optimal choices in planning for its goal. This is a strong assumption and we consider the implications of weakening it shortly. Even under this assumption, identifying goals is hard. A further problem is that goals can be any subset of facts that is achievable from the initial state so, as we observe actions, each new state could be the goal — the path taken to reach it will be the shortest path (if there were a shorter path then it would contradict our assumption that the agent is executing an optimal plan) and, therefore, it will be consistent with the observations that this is precisely what the agent intended to achieve.

In general, there are many goal sets consistent with a sequence of observed actions, ranging from the possibility that the most recent state was in fact the goal state to the possibility that there are many goals towards which the agent has not yet even begun to act. However, these possibilities are not all equally likely: in most domains there is a clear bias towards certain kinds of goals. This motivates the following definition:

Definition 2. *Goal Hypothesis and Goal Hypothesis Space*
Given a goal recognition problem base, G, with facts F, a goal hypothesis for G is a probability distribution over subsets of F reachable from the initial state using actions in G. The goal hypothesis space for G, $\mathbb{H}$, is the set of all such goal hypotheses for G.

Note that as a special case a goal hypothesis might assign equal non-zero probabilities to some subset of reachable sets of facts and zero to all others (that is, a uniform distribution over a subset of the candidate goals). We will refer to this case as the uniform goal hypothesis over this subset of goals.

Definition 3. *Goal Recognition Problem*
A goal recognition problem is a triple, $\langle G, H_I, (o_1, o_2, ..., o_n)\rangle$, where G is a goal recognition problem base, H_I is an initial goal hypothesis and $(o_1, ..., o_n)$ is the sequence of actions observed one-by-one during the problem.

Each observation in a goal recognition problem updates the hypothesis space, so that candidate goals that are further away from the new state than the previous state are assigned an updated probability of 0, while the remaining probability mass is re-normalised across the other states. Nothing observable in the sequence can lead us to modify the relative probabilities of goals that have a common shortest path following the observations made so far. It is interesting to note that actions are transitions between goal hypotheses in an analogous way to their behaviour as transitions between states.

Unfortunately, a goal hypothesis is potentially exponentially large in the size of the set of facts for the underlying planning problem. This is generally far too large to make it possible to represent goal hypotheses explicitly. However, in the work of Ramírez and Geffner [22], the initial goal hypothesis *is* described explicitly, by enumeration (although in their work they do not refer to probabilities). Their work can be interpreted as offering a way to handle the special case of uniform goal hypotheses, but restricted to small (that is, explicitly enumerated) subsets of candidate goal sets.

Explicit representation of $\mathbb{H}$ for anything other than trivial problems is impossible, due to its exponential size. We therefore introduce an approximation of the space which is tractable, but at the price that we cannot accurately represent all possible goal hypotheses.

Definition 4. *Approximate Goal Hypothesis*
An nth order approximation to a goal hypothesis, H, is a goal hypothesis, $\hat{H}$, where $\hat{H}(f) = H(f)$ when $|f| \leqslant n$ and, $\hat{H}(f) = \min_{x\in f} \hat{H}(f \setminus \{x\}) \cdot \hat{H}(\{x\})/N$, where N is an appropriate normalising factor to ensure that $\hat{H}$ is a probability distribution.

An approximate goal hypothesis is not necessarily a member of the same goal hypothesis space as the goal hypothesis it approximates, because the approximation can assign non-zero probabilities to unreachable sets of facts. Identifying unreachable sets is as hard as planning, so allowing these sets to be assigned non-zero values is a useful efficiency measure. The method by which probabilities are combined in the recursive extension of the approximation to the whole space of possible goals is somewhat arbitrary and alternative approximations are certainly possible. In our current work we only consider 1st order approximations, so the probability of sets of facts is the product of the probabilities of the individual facts they include. This is equivalent to assuming that the individual goals have independent probabilities of appearing. Although 1st order approximations are poor in domains where goals are strongly correlated, in many domains we see goals falling into independent selections of states of a collection of objects (such as packages in a delivery domain).

This independence assumption clearly does not hold for all domains, for example BLOCKSWORLD problems often have the same numbers of goal and initial-state literals. We currently focus on problems which do exhibit this property, although we also consider the performance of the approximation on other benchmark domains.

Within the framework we have now defined, it is apparent that each successive observation implies an update of the current goal hypothesis reducing the probabilities of reachable facts that are subsets of those states which are now no closer in the state than the prior state. However, it is impractical to identify exactly which states these are. Furthermore, the assumption that the agent that is being observed has the capability to identify the shortest path to its goal, without error, is unreliable. For this reason, we work with 1st order approximations and update by reducing the probability associated with *facts* that get further away following observed actions and increasing the probabilities of *facts* that get closer.

4 RECOGNITION WITHOUT LIBRARIES

AUTOGRAPH performs goal recognition in four stages: *Analysis*, wherein the problem is instantiated and analysed to reveal useful aspects of the domain; *Observation*, in which a single, ordered action is fed into the recogniser and the current state updated to reflect its effects; *Intermediate Hypothesis Generation*, in which a single hypothesis is produced after each observation[2]; and, lastly, a *Final Hypothesis* is generated once the plan is known to have finished.

4.1 Analysis

Domain analysis can provide rich information to aid subsequent search [6, 10, 11], lowering search time and shortening plan length. We apply relevant prior research to GR and develop new techniques that allow the recogniser to make more informed hypotheses.

4.1.1 Problem Representation

We use domains encoded in PDDL2.1 [7] and then apply Helmert's translator [12] to translate these into a SAS+ formalism. Two key products of this translation process are Domain Transition Graphs (DTGs) and a Causal Graph (CG), both of which encode aspects of the original PDDL problem in another form. We use both the PDDL and SAS+ representations of the problem during analysis, as they can each reveal aspects of the domain that aid in recognition. The Causal Graph reveals how objects influence others within the domain through actions: of particular interest are the *leaf nodes*, corresponding to objects with no influence on others.

Definition 5. *CG Leaves*
*A node, v, in the causal graph with $|v_{out}| = 0$ is a **leaf variable** and any fact which contains a leaf variable in its parameters is a **leaf proposition**.*

Should a causal graph contain leaf nodes, any leaf proposition that is **not** true in the initial state can be seen as a likely goal, as it can play no role other than to be altered.

Modern planners typically use a *grounded problem* that contains **all** possible fact and action combinations, often including some that are unreachable (e.g. `(on crate1 crate1)`). We filter this set to include only reachable facts which can then be further analysed to reveal certain domain characteristics that aid in recognition.

By using an action-centred model to define domains we lose the knowledge present in more structured hierarchical models, but note that such information can often be extrapolated through domain analysis. It is also the case that, while our first-order approximation will reduce the size of the goal-space, it often makes determining the true goal state of an object difficult. Where object goal states are modelled by properties drawn from several sets, the first-order approximation will not explicitly link these properties. However, plans achieving such collections will show correlated increases in probabilities for all of the goal properties, so the linkage is implicit in the treatment of the updates.

4.1.2 Predicate Partitioning

Geib [8] proposed the concept of *plan heads* in Plan Recognition as a way to highlight important plan actions and allowing lazy commitment to plans, resulting in faster runtimes. We adapt this idea for GR through the concept of *predicate partitioning*. By automatically classifying propositions into mutually-exclusive sets it can often become clear which are more or less likely to be goals. For example, in a standard Logistics problem it is unlikely that the goal will be to simply have a package inside a truck and far more likely that it must be delivered to a warehouse. Facts can be placed in the following sets through analysis of the two domain representations, which can then be applied to the initial probability distribution.

Definition 6. *Predicate Partitions – A fact f is:*

1. ***strictly activating** iff $f \in I$ and $\forall a \in A$, $f \notin \mathit{eff}_a$ and $\exists a \in A$, $f \in pre_a$, where A is the set of grounded actions and $\mathit{eff}_a = add_a \cup del_a$. Strictly activating (SA) facts are often referred to as* static. *These facts can never be removed from the state and therefore extremely unlikely to be goals.*
2. ***unstably activating** iff $f \in I$ and $\forall a \in A$, $f \notin add_a$ and $\exists a \in A$, $f \in pre_a$ and $\exists a \in A$, $f \in del_a$. Unstably Activating facts differ from SA facts in that they can be deleted by at least one action, but once removed from the current state, cannot be re-added. Once deleted they can be removed from future hypotheses.*
3. ***strictly terminal** iff $\exists a \in A$, $f \in add_a$ and $\forall a \in A$, $f \notin pre_a, f \notin del_a$. These facts are assigned a high initial probability. Once added to the current state, they will not be removed, meaning they must appear in the final state.*
4. ***unstably terminal** iff $\exists a \in A$, $f \in add_a$ and $\forall a \in A$, $f \notin pre_a$, but $\exists a \in A$, $f \in del_a$. Unlike strictly terminal facts, these can be removed once they have been added, but they are never used as preconditions to any actions. They may simply be an uninteresting side-effect of an action, or involved in a mutex-relation.*
5. ***waypoint** iff f has predicate symbol $pred_f$ and $\forall q \in dtg(f)_{out}$, $|dtg(f)_{out}| \geqslant 2$, and $\bigcup_{i=1}^{n} pred_{q_i} = pred_f$, where $dtg(f)_{out}$ is the set of DTG vertices to which f is connected by at least two outgoing edges. It is common for problems to involve transforming objects through a chain of related states, all defined by the same predicate. The facts located within this predicate-chain (excluding end-points) are waypoint facts, and are assigned a low initial probability.*
6. ***transient** iff the predicate symbol for f is $pred_f$ and $\forall q \in dtg(f)_{out}, \bigcup_{i=1}^{n} pred_{q_i} = pred_{q_1}$ and $pred_{q_1} \neq pred_f$, where $dtg(f)_{out}$ is the set of DTG vertices to which f is connected by at least two outgoing edges. While waypoints form chains of facts sharing the same predicate, transient facts are intermediate values in a transition that use a different predicate. Furthermore, objects entering the transient state must return to a state with the same predicate as the one they originally left. It is generally unlikely that the goal will be to leave the object within this intermediate state so we assign them a low probability in the initial-probability distribution.*
7. ***binary** iff $\forall dtg \in DTG, |dtg(f)| = 2$. As a special case, facts that have a DTG of size 1 are also considered binary, by including the negated literal. Binary facts are assigned low initial probabilities as it is difficult to assess which of them might be relevant to the goal.*

In addition to these sets, a further *neutral* set is defined, containing all facts that have not been partitioned into one of the above sets.

The population of the various partitions is dependent on the domain being analysed. For example the ROVERS domain populates 5 partitions, while others such as ZENOTRAVEL largely categorise facts in the *waypoint* and *transient* sets. Overall results of this partitioning process on the testing domains can be seen in Table 1[3]. The populations of these partitions are used during construction of the *initial-probability distribution*.

[2] If the plan has further actions to be observed, steps 2 and 3 are repeated until this is no longer the case.

[3] Note that *strictly activating* facts do not appear in any of the test domains, as they are filtered during grounding.

Problem	**T**	**W**	**UT**	**ST**	**UA**	**SA**	**B**	**N**
Driverlog	3127	2005	0	0	0	0	67	741
Depots	3773	238	0	0	0	0	423	1940
Rovers	0	3986	0	592	1001	0	736	11159
Zenotravel	3350	1091	0	0	0	0	0	77
Storage	2417	683	456	0	0	0	327	493

Table 1: Abbreviated results indicating partition totals for the problem files associated with each domain.

4.1.3 Landmarks

The use of *landmarks* in heuristic search has shown that they can be a powerful guide through the search-space [21, 24]. We detect landmarks for all $f \in \mathcal{H}$, where $\mathcal{H}$ is the set of all facts within the goal-space, and store these in lists of the form $(Lm_1, Lm_2...Lm_n, f)$. These landmark-lists show which facts are "stepping-stones" to the final goal. For instance, if `(in truck1 package3)` is added to the current state after an observation and that same fact is in two landmark lists, it is sensible to increase the probability of propositions which follow this landmark as being goals – or a the very least not lower their probability. This is covered in further detail during the *Execution* section.

4.1.4 Unhelpful Facts

While Helmert's SAS+ translation also approximates the set of all *reachable* facts, it is likely that some will never appear as a goal. We begin to reduce the set of facts by first observing that it is extremely unlikely that a problem will be considered a planning problem if its goals can be achieved in a single step, since this could be achieved by purely greedy action selection. Therefore, any action applicable in the initial state is considered *unhelpful* and its effects are assigned negligible probability in the initial hypothesis. Additionally, if the domain contains *strictly-terminal* facts, we assign negligible probability to the preconditions of any action which achieves them by reasoning that the enabling conditions for achievement of a *strictly-terminal* fact are very unlikely to be goals instead of the terminal itself.

Definition 7. *Unhelpful Facts*
Given an initial state I, the set of ***unhelpful effects*** *is equivalent to $\bigcup_{m=1}^{|A_I|} add_{a_m}$ where A_I is the set of actions applicable in I. Furthermore, if a domain exhibits strictly-terminal facts F_{sTerm}, the set of* ***unhelpful preconditions*** *is equivalent to $\bigcup_{n=1}^{|A_{ST}|} pre_{a_n}$ where A_{ST} is the the set of all actions that achieve any member of F_{sTerm}.*

4.1.5 Initial Probability Distribution

Once the analysis phase has been completed, each fact f in the approximate hypothesis space $\mathcal{H}$ can be assigned an initial probability. This figure is dependent on which, if any, of the previous domain analysis criteria the fact has met.

The values assigned during this phase are selected directly according to which partition facts belong to among those described above. For example, *strictly terminal* and *unstably terminal* facts are given very high values because they only exist to be achieved, i.e. they do not appear as a precondition to any action. Conversely, *strictly activating* facts are given a probability of 0 as they will remain true throughout execution, making their appearance in the goal futile. *Waypoint*, *transient* and *binary* facts must be treated with more caution, as it is conceivable that they could be a goal, but more likely to be used as a means of achieving other goal propositions. Furthermore, *leaf propositions* which are not true in the initial state are all assigned the same value, as it is equally likely that the *leaf variable* contained within them could appear in any one of them in the goal state. The values selected for each partition are somewhat arbitrary and we have not yet explored the effect of a wide range of alternative assignments. Our initial experiments suggest that the system is not particularly sensitive to the precise choice of values.

4.2 Execution

Once the domain has been analysed and the initial goal-space populated, plan observation can begin. After each observation we record the *heuristic estimate* to each fact in the *approximate hypothesis space*.

By observing the estimated distance to each fact after action observations, it is possible to determine those which are being moved towards and away from. Each fact which has a lower heuristic estimate at time t than it did at $t - 1$ has its probability of being a goal increased, while those which now have a larger estimate have their probability set to 0. Facts whose estimate remains unchanged do not have their probability updated as they may be goals which have been achieved at time t.

4.2.1 Landmark-Linked Facts

The above probability increments and decrements apply to all propositions in the goal space, with the exception of those which appear as *landmarks* of other goals. If a fact L_f is classified as a landmark of another fact and is getting closer, then it is reasonable to assume any facts ordered *after* L_f are more likely to appear than other facts.

Landmarks which succeed L_f have their probabilities increased as it is possible they may be the true goal. The increment value of these successors is lower than a standard fact increment, with this decreasing linearly over any successor landmarks until 0 or the end of the list is reached. However, if the predecessor of L_f is being moved away from then L_f and successor landmarks do not have their probability modified. This is due to the possibility of search "turning around" towards L_n once L_{n-1} is achieved.

4.3 Hypothesis Generation

By using a 1st order approximation of a goal hypothesis we rely on goal sets being small. However, it would be naïve to assume that all domains only contain a single literal as their goal. We therefore construct an *intermediate greedy hypothesis* h_i from the approximate goal hypothesis constructed at each timestep i, representing the single most likely goal of the agent being observed.

To produce this set, facts are considered in mutually exclusive clusters (the sets that make up the nodes in a single DTG). The fact with highest probability within each cluster is selected, provided it has probability higher than a specified threshold (this eliminates highly unlikely candidates from the set).

Definition 8. *Greedy-Hypothesis*
Given an approximate hypothesis space $\mathcal{H}$, a ***greedy-hypothesis*** *h_{gr} is the set of facts with the highest probabilities above a base threshold, T_{min}, with ties broken randomly. If a fact f is chosen, then all facts that are mutex with it will not be added to h_{gr}.*

4.4 Final Hypothesis

Once the plan is known to have terminated and the final state is known, a more accurate final hypothesis can be produced. This is simpler than generating an *intermediate hypothesis* since $G \subseteq S_n$, and the state is certainly mutex-free. Along with the final probability distribution, this can produce a very accurate goal hypothesis without considering fact probabilities.

5 EVALUATION

We now present empirical results of several tests performed on the techniques presented previously. While others have previously expressed a desire for plan and goal recognition to have a standard evaluation method [4], there is still no agreement on standard benchmarks. Therefore, we have used classical planning benchmarks as an alternative. The system is evaluated using *precision and recall*, a technique used to score database document-retrieval which has also previously been applied to a GR context [2], where the number of required facts in each hypothesis is the *precision* and the number of correct facts is the *recall*.

The test system is written in Java and makes extensive use of a Java implementation of the FF planner [5]. SAS+ translation is performed using Helmert's standard Python scripts [12], which is then converted into a Java representation. Tests were conducted in Ubuntu 9.10 on a quad core 2.8GHz Intel i5 with 4GB of RAM using the latest Java Virtual Machine (1.6.0_14), and were given as much time as necessary to complete each stage of the recognition process.

The domains used in testing are taken from the 2002 and 2006 International Planning Competitions – specifically the propositional/STRIPS versions of DRIVERLOG, DEPOTS and ZENOTRAVEL, ROVERS and STORAGE, along with their best-known-solutions.

All of the domains have 20-40 associated problem files[4] which become increasingly difficult for planners to solve. Harder problems have much larger goal-spaces than trivial problems, which means that the recogniser has a much wider *hypothesis space* to work with. However, in compliance with our expectations, the number of facts in the goal state of these problems does not increase at the same rate as the number of *grounded facts*.

We have tested the system using the *Max* (h_{max}), *FF* (h_{ff}) and *Causal Graph* (h_{cg}) heuristics [3, 13, 23] in order to determine how this choice affects performance. We note that in testing with these heuristics in combination with solutions produced by planners which also use the heuristic there is the possibility of *heuristic bias*, but that this should not adversely affect the results.

5.1 Intermediate Hypothesis Results

The results of precision versus recall for *intermediate hypotheses* over all domains and heuristics can be seen in Table 2. Also included are the average precision and recall over all problems using h_{ff} at various timepoints. These latter results show *heuristic convergence* as precision and recall increase at over the course of plan execution.

The process of *heuristic-convergence* is more clearly visible in Figure 1 which contains the results of a typical ZENOTRAVEL problem. All three heuristics move towards the true goal, but with occasional dips in accuracy. This is a result of all three heuristics chosen being *inadmissible*. By being able to both over and under-estimate the distance to a goal, facts may be incorrectly classed as becoming further away after an observation. In the future it may be of use to apply an *admissible* heuristic which would guarantee never to over-estimate the distance, and thus possibly remove these fluctuations. Note that h_{cg} provides the fewest drops in accuracy, but not the quickest convergence rate.

Perhaps of most interest is that there is no clear leader in terms of heuristic chosen to generate estimates. While h_{cg} has the highest overall P+R results, this is primarily caused by the results of ROVERS, which contains more tests than other domains. The normalised results in Table 2 show the difference between this and h_{ff}

[4] All problems were tested with the exceptions of DEPOTS 21 and 22, ROVERS 37-40, STORAGE 24-30 which could not be grounded correctly.

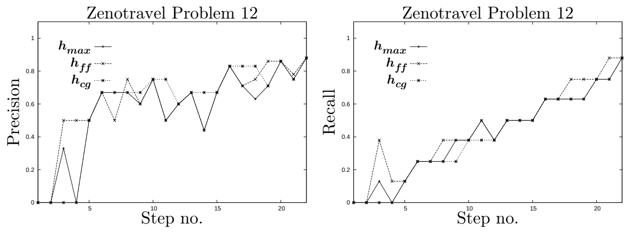

Figure 1: Intermediate P+R results for a single problem instance in ZENOTRAVEL.

is minor, while Figure 1 shows how close the estimates are on an individual problem.

5.2 Final Hypothesis Results

Once the plan being observed is known to have finished, the *final hypothesis* can be generated. Table 3 shows the total normalised P+R values for the final hypotheses in each problem. We note that the seeming indifference of heuristic choice is further reinforced by these results, as all three heuristics produce identical final P+R results.

With the exception of DEPOTS and STORAGE, the recogniser produces highly accurate hypotheses for all problems in terms of both precision and recall. ROVERS shows particularly accurate results due to the presence of *strictly-terminal* facts, which produces a perfect score for recall, and 95% average for precision.

In the case of DEPOTS and STORAGE, precision scores average only 52% and 32% respectively. This is caused by the large number of facts which become true during execution of a typical plan, which along with a small goal set combine to form a large hypothesis with extraneous facts. For instance, the location of certain trucks is often not a required goal in DEPOTS, but will be put forward as a goal because trucks will stop moving once the last package has been delivered to its destination.

Domain	Depots	Driverlog	Rovers	Zenotravel	Storage
Precision	0.52	0.94	0.96	0.86	0.32
Recall	0.93	0.73	1	0.7	0.95

Table 3: Normalised total values for precision and recall values associated with the final hypothesis for each domain

5.2.1 *Accuracy of Initial Hypothesis After Final Observation*

In the *Analysis* section we described the generation of an *initial probability distribution* using only domain analysis. At the time it this generated, it is unlikely that construction of a *greedy-hypothesis* would produce an accurate result as no observations have been seen. However, if this initial, unmodified probability distribution is combined with the final state it can often produce an accurate hypothesis. Table 4 shows the difference in overall P+R when this is compared with the true final hypothesis. These results show that, in the case of DEPOTS, precision can be increased by over 40% whilst also increasing recall, while ZENOTRAVEL also shows this to a lesser extent. Moreover, the presence of *terminal* facts in STORAGE and ROVERS shows that it is possible to achieve almost 100% accuracy on both precision and recall without modifying the *initial probability distribution*. Only DRIVERLOG shows a reduction of both precision and recall.

Improvement	Depots	Driverlog	Rovers	Zenotravel	Storage
Precision	40.57%	-3.95%	4.10%	4.77%	49.63%
Recall	6.73%	-15.95%	0.00%	13.91%	10.05%

Table 4: Differences in P+R if the initial probability distribution is used during generation of a final hypothesis.

For some domains the risk of lowering of P+R is not preferable,

Domain	h_{max}	h_{ff}	h_{cg}	Domain	h_{ff} 25%	h_{ff} 50%	h_{ff} 75%	h_{ff} 100%
Depots	**0.22 / 0.3**	0.22 / 0.28	0.22 / 0.28	Depots	0.15 / 0.07	0.2 / 0.21	0.34 / 0.5	0.52 / 0.93
Driverlog	0.42 / 0.32	**0.47 / 0.32**	**0.47 / 0.32**	Driverlog	0.43 / 0.12	0.56 / 0.29	0.72 / 0.5	0.94 / 0.73
Rovers	0.78 / 0.54	0.82 / 0.5	**0.86 / 0.56**	Rovers	0.62 / 0.22	0.81 / 0.42	0.93 / 0.66	0.96 / 1
Zenotravel	0.46 / 0.32	**0.49 / 0.33**	0.48 / 0.31	Zenotravel	0.3 / 0.1	0.42 / 0.26	0.61 / 0.48	0.81 / 0.7
Storage	**0.19 / 0.42**	**0.22 / 0.39**	**0.22 / 0.39**	Storage	0.17 / 0.14	0.29 / 0.49	0.27 / 0.64	0.3 / 0.91
Total	0.54 / 0.43	0.58 / 0.4	**0.6 / 0.43**	Average	0.33 / 0.13	0.46 / 0.33	0.57 / 0.56	0.7 / 0.85

Table 2: The total normalised intermediate precision and recall results for each heuristic, and the average precision and recall for h_{ff} over all problems at 25%, 50%, 75% and 100% plan completion.

but in a situation where computing resources are limited and the domain is known to exhibit this property, it may be preferable to accept a small loss in recall and save any processing-time related to the updating of probabilities after each observation.

6 CONCLUSIONS AND FUTURE WORK

We have presented AUTOGRAPH, a new method of tackling Goal Recognition by applying Planning technology. The approach and empirical evidence presented has successfully shown that libraries are not required to achieve online recognition of an agent's activities.

The work presented offers a novel approach to the problem in the form of *heuristic estimation*, as well as several new methods of refining valid goal facts. Perhaps most importantly it offers a viable solution to the problem of offline library construction and allows **any** domain to be recognised without prior analysis.

6.1 Strengths

AUTOGRAPH demonstrates an effective and efficient performance in goal recognition. It is **complete** in the sense that it can construct a hypothesis from *any* conjunction of literals within $\mathcal{H}$. The system is **scalable** because it is based on a 1st order approximation of the true goal hypothesis, meaning that the hypothesis grows only linearly with the size of the grounded problem. Finally, it is **domain-independent** because it only relies on the use of a standard problem definition schema and use of generic heuristics and algorithms.

Several new and interesting aspects of GR have also been observed. For example, the ability to accurately predict goals purely from initial domain analysis and the current state is an interesting avenue of research which could make the system more applicable in low-resource environments.

6.2 Limitations

A drawback of the system is the inability to know if a hypothesis is valid, due to the problem of detecting all mutually-exclusive propositions. The detection of these mutexes is an NP-Hard problem, meaning that other methods must be used to estimate propositions which cannot exist together in the same state. Undetectable mutexes are a domain-specific feature, with only DEPOTS showing this trait from the set of test domains. Future work will explore the *approximation* of mutex information by recording facts which never appear together during intermediate plan-states.

The current linear convergence rate of the recogniser is to be expected from the heuristic estimation process, but a faster convergence rate would obviously be preferable. One method of increasing convergence rates could be to rule out any facts which cannot be reached within n steps, where $n > |P|$. However, in order to do this the problem of plan-length estimation would need to be solved first, along with the detection of accurate goal-conjunctions. Additionally, automating the process of selecting initial probabilities for each partition and during updates on a domain-by-domain basis using a system such as Hoos *et al* [16] would reveal the optimal set of values for generating fast and accurate hypotheses.

ACKNOWLEDGEMENTS

Thanks go to Bram Ridder for access to his Java SAS+ translator and Causal Graph heuristic code.

REFERENCES

[1] Nate Blaylock and James F. Allen, 'Corpus-based, statistical goal recognition', in *Proc. Int. Joint Conf. AI*, pp. 1303–1308, (2003).
[2] Nate Blaylock and James F. Allen, 'Fast hierarchical goal schema recognition', in *Proc. Nat. Conf. on AI (AAAI)*, pp. 796–801, (2006).
[3] Blai Bonet and Hector Geffner, 'Planning as heuristic search', *J. AI Res.*, **129**(1-2), 5–33, (2001).
[4] Sandra Carberry, 'Techniques for plan recognition', *User Modeling and User-Adapted Interaction*, **11**(1-2), 31–48, (2001).
[5] Andrew Coles, Maria Fox, Derek Long, and Amanda Smith, 'Teaching forward-chaining planning with JavaFF', in *Colloquium on AI Education, 23rd AAAI Conf. on AI*, (2008).
[6] Maria Fox and Derek Long, 'The automatic inference of state invariants in TIM', *J. AI Res.*, **9**, 367–421, (1998).
[7] Maria Fox and Derek Long, 'PDDL2.1: An extension to PDDL for expressing temporal planning domains', *J. AI Res.*, **20**, 61–124, (2003).
[8] Christopher W. Geib, 'Delaying commitment in plan recognition using combinatory categorial grammars', in *Proc. Int. Joint Conf. on AI*, pp. 1702–1707, (2009).
[9] Peter Gorniak and Deb Roy, 'Probabilistic grounding of situated speech using plan recognition and reference resolution', in *Proc. 7th Int. Conf. on Multimodal Interfaces*, pp. 138–143, (2005).
[10] Malte Helmert, 'A planning heuristic based on causal graph analysis', in *Proc. Int. Conf. on Automated Planning and Scheduling*, pp. 161–170, (2004).
[11] Malte Helmert, 'The fast downward planning system', *J. AI Res.*, **26**, 191–246, (2006).
[12] Malte Helmert, 'Concise finite-domain representations for PDDL planning tasks', *Artif. Intell.*, **173**(5-6), 503–535, (2009).
[13] Jörg Hoffmann and Bernhard Nebel, 'The FF planning system: Fast plan generation through heuristic search', *J. AI Res.*, **14**, 253–302, (2001).
[14] Jun Hong, 'Goal recognition through goal graph analysis', *J. AI Res.*, **15**, 1–30, (2001).
[15] Alexander Huntemann, Eric Demeester, Hendrik van Brussel, and Marnix Nuttin, 'Can Bayes help disabled users? A Bayesian approach to plan recognition and shared control for steering an electrical wheelchair', in *Proc. ACM/IEEE Human-Robot Interaction Conf.*, pp. 67–70, (2008).
[16] Frank Hutter, Domagoj Babic, Holger H. Hoos, and Alan J. Hu, 'Boosting verification by automatic tuning of decision procedures', in *FMCAD '07: Proc. Formal Methods in CAD*, pp. 27–34, (2007).
[17] Henry A. Kautz, *A formal theory of plan recognition*, Ph.D. dissertation, University of Rochester, 1987.
[18] Neal Lesh, 'Adaptive goal recognition', in *Proc. Int. Joint Conf. on AI*, pp. 1208–1214, (1997).
[19] Neal Lesh and Oren Etzioni, 'A sound and fast goal recognizer', in *Proc. Int. Joint Conf. on AI*, pp. 1704–1710, (1995).
[20] Bradford Mott, Sunyoung Lee, and James Lester, 'Probabilistic goal recognition in interactive narrative environments', in *Proc. Nat. Conf. on AI (AAAI)*, pp. 187–192, (2006).
[21] Julie Porteous, Laura Sebastia, and Jörg Hoffmann, 'On the extraction, ordering, and usage of landmarks in planning', in *Proc, European Conference on Artificial Intelligence*, pp. 37–48, (2001).
[22] Miquel Ramírez and Hector Geffner, 'Plan recognition as planning', in *Proc. Int. Joint Conf. on AI*, pp. 1778–1783, (2009).
[23] Silvia Richter, Malte Helmert, and Matthias Westphal, 'Landmarks revisited', in *AAAI*, pp. 975–982, (2008).
[24] Matthias Westphal and Silvia Richter. The LAMA planner. using landmark counting in heuristic search, 2008. Short paper for IPC 2008.