

An LP-Based Approach for Goal Recognition as Planning

Luísa R. de A. Santos¹, Felipe Meneguzzi², Ramon Fraga Pereira³, André G. Pereira¹

¹Federal University of Rio Grande do Sul, Brazil

²Pontifical Catholic University of Rio Grande do Sul, Brazil

³Sapienza University of Rome, Italy

¹{lrasantos, agpereira}@inf.ufrgs.br

²felipe.meneguzzi@pucrs.br

³pereira@diag.uniroma1.it

Abstract

Goal recognition aims to recognize the set of candidate goals that are compatible with the observed behavior of an agent. In this paper, we develop a method based on the *operator-counting framework* that efficiently computes solutions that satisfy the observations and uses the information generated to solve goal recognition tasks. Our method reasons explicitly about both partial and noisy observations: estimating uncertainty for the former, and satisfying observations given the unreliability of the sensor for the latter. We evaluate our approach empirically over a large data set, analyzing its components on how each can impact the quality of the solutions. In general, our approach is superior to previous methods in terms of agreement ratio, accuracy, and spread. Finally, our approach paves the way for new research on *combinatorial optimization* to solve goal recognition tasks.

Introduction

Goal recognition as planning consists of inferring the set of compatible goals from a set of goal candidates, given a planning task without a goal, and a sequence of observations. A solution for a goal recognition task is a subset of goal candidates that are compatible with the sequence of observations. A plan for the planning task with the reference goal, part of the set of goal candidates, generates the sequence of observations. This sequence may be partial, containing any number of observations from the plan. Existing methods on goal recognition try to cope with three main classes of observation sequences: optimal (Ramírez and Geffner 2009), sub-optimal (Ramírez and Geffner 2010), and noisy (Sohrabi, Riabov, and Udrea 2016). Since approaches to goal recognition as planning often employ standard planning technology to solve goal recognition tasks, many of them can benefit from improvements in the underlying planning technology (Ramírez and Geffner 2009; E-Martín, R.-Moreno, and Smith 2015; Pereira, Oren, and Meneguzzi 2017; Harman and Simoens 2020).

Recent developments in planning include heuristics based on the *operator-counting framework*, which combines the

information of admissible heuristic functions through an *integer program* (IP) (Pommerening et al. 2014). These heuristics provide constraints that must be satisfied by every plan of the planning task. In general, the objective value of the *linear program* (LP), a linear relaxation of the integer program, is used as heuristic function to guide the search. A major advantage of this framework is that it enables to reason and to manipulate the information of the heuristics directly.

We develop an LP-based approach to solve goal recognition tasks, including five main contributions. First, we modify the operator-counting framework to efficiently compute solutions that satisfy the counts of observations of a goal recognition task. We also use this framework to estimate the cost of an optimal plan for each goal candidate in the task. Then, we use the information generated to solve the goal recognition task. Second, we show how to contrast the objective value of the modified linear program and the length of the sequence of observations to estimate the uncertainty of the decision of our approach which we use to improve our solution. Third, we develop an approach to explicitly address noisy observations. Given the unreliability of the sensor of observations, we create an integer program that aims to automatically ignore noisy observations when computing solutions. Fourth, we show that higher heuristic values from lower bound heuristics for the reference goal predict the quality of our solution. Finally, we modify the previous benchmarks to compare goal recognition methods by agreement ratio, showing that ours overcomes the state of the art.

Planning Task and Operator-Counting Framework

An *SAS⁺ planning task* is a tuple $\Pi = \langle \mathcal{V}, \mathcal{O}, s_0, s^*, \text{cost} \rangle$, where $\mathcal{V}$ is a set of *variables*, $\mathcal{O}$ is a set of *operators*, s_0 is an *initial state*, s^* is a *goal condition*, and cost a cost function. Each variable $v \in \mathcal{V}$ has a finite domain $D(v)$. A *state* is a complete assignment, a *partial state* is a partial assignment of the variables over $\mathcal{V}$, $\text{vars}(s)$ is the set of variables in a (partial) state s , and $s[v]$ is the value of variable v in a (partial) state s . The initial state s_0 is a state, and the goal condition s^* is a partial state. A state s is consistent with a (partial) state s' if $s[v] = s'[v]$ for all $v \in \text{vars}(s')$. Each operator $o \in \mathcal{O}$ is pair of partial states $\langle \text{pre}(o), \text{post}(o) \rangle$

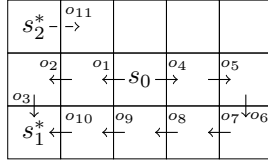


Figure 1: A goal recognition task example.

and an operator o is applicable to a state s if s is consistent with $\text{pre}(o)$. Applying o to s generates a new state s' such that $s'[v] = \text{post}(o)[v]$ for $\text{vars}(\text{post}(o))$ and for the remaining variables $s'[v] = s[v]$. Function $\text{cost} : \mathcal{O} \rightarrow \mathbb{Z}_0^+$ assigns a non-negative cost to each operator $o \in \mathcal{O}$ – in this paper all operators have unit cost. An s -plan π is a sequence of operators $\langle o_1, \dots, o_n \rangle$ such that there exists a sequence of states $\langle s_1 = s, \dots, s_{n+1} \rangle$ where o_i is applicable to s_i and produces state s_{i+1} , and s_{n+1} is consistent with s^* . The cost of a s -plan π is defined as $\text{cost}(\pi) = \sum_{o \in \pi} \text{cost}(o)$. An s_0 -plan or a *plan* is a solution to a planning task and is *optimal* if its cost is minimal. Figure 1 illustrates our running example where the agent performs cardinal movements and starts at s_0 . An optimal plan that reaches s_1^* for this task is $\pi = \langle o_1, o_2, o_3 \rangle$.

Definition 1 (Operator-Counting Constraint). Let Π be a planning task with operators $\mathcal{O}$, and let s be one of its states. Let $\mathcal{Y}$ be a set of real-valued and integer variables, including an operator-counting non-negative integer variable Y_o for each operator $o \in \mathcal{O}$. A set of linear inequalities over $\mathcal{Y}$ is an operator-counting constraint for s if for every valid s -plan π , there exists a solution with $Y_o = \text{occur}_\pi(o)$ for all $o \in \mathcal{O}$ – where $\text{occur}_\pi(o)$ is the number of occurrences of operator o in the s -plan π .

In the example $Y_{o_3} + Y_{o_{10}} \geq 1$ is an operator-counting constraint (and a landmark constraint) for goal s_1^* because the agent must use one of these operators to reach s_1^* .

Definition 2 (Operator-Counting IP/LP Heuristic). The operator-counting integer program IP^C for a set of operator-counting constraints C for state s is

$$\begin{aligned}
 & \text{minimize } \sum_{o \in \mathcal{O}} \text{cost}(o) Y_o \\
 & \text{subject to } C, \\
 & Y_o \in \mathbb{Z}_0^+.
 \end{aligned}$$

The IP heuristic h^{IP} is the objective value of IP^C , and the LP heuristic h is the objective value of its linear relaxation. If the IP or LP is infeasible, the heuristic estimate is ∞ .

Goal Recognition as Planning

We formally define the task of *goal recognition as planning* as a tuple $\langle \Pi_P, \Gamma, \Omega \rangle$, where Π_P is a planning task without a goal condition, Γ is a set of goal candidates, and Ω is a sequence of observations. Observation $\vec{o}$ corresponds to operator o . For readability, we abuse notation and equate operators to observations throughout the paper when convenient.

Definition 3 (Observation Compliance). Let $\pi = \langle o_1, \dots, o_n \rangle$ be a plan for a planning task Π and $\Omega = \langle \vec{o}_1, \dots, \vec{o}_m \rangle$ a sequence of observations. Plan π complies with Ω if a monotonic function $f : [1, m] \mapsto [1, n]$ exists that maps all operator indexes in Ω to indexes in π , such that $\vec{o}_i = o_{f(i)}$.

We define three classes of sequences of observations: *optimal* and *sub-optimal* (Definition 4) observations, and *noisy optimal/sub-optimal* observations (Definition 5).

Definition 4 (Sequence of Observations). Let $\pi = \langle o_1, \dots, o_n \rangle$ be a plan for the planning task Π . Then, a sequence of observations Ω is a sequence of operators extracted from the plan π maintaining their relative order. The sequence may be partial, containing any number of operators from the plan π . An optimal sequence of observations is extracted from an optimal plan and a sub-optimal sequence of observations is extracted from a sub-optimal plan. An optimal/sub-optimal observation is part of an optimal/sub-optimal sequence of observations.

Definition 5 (Noisy Observations). A noisy sequence of observations Ω is a sequence observations extracted from π that is augmented with at least one observation $\vec{o} \in \Omega$ which is inserted in any position of the extracted sequence.

We extend the standard definition from Ramírez and Geffner (2009) of an *exact solution set* for a goal recognition task to also consider sub-optimal observation sequences (Definition 6) and call it *reference solution set*. We define the *reference solution set* as a subset of the goal candidates such that there exists a complying plan as sub-optimal as or less than the plan that generated the observations for the reference goal.

Definition 6 (Reference Solution Set). Let $\langle \Pi_P, \Gamma, \Omega \rangle$ be a goal recognition task and Π a planning task with the goal condition $s^* \in \Gamma$ (the reference goal). Let π^* be an optimal plan for Π , and let π be a plan for Π from which Ω is extracted. Let $h_\Omega^*(s_0, s_i^*)$ be the cost of an optimal plan for Π restricted to the set of plans that comply with Ω , and $h^*(s_0, s_i^*)$ be the cost for an optimal plan for Π , both with $s_i^* \in \Gamma$. $h_\Omega^*(s_0, s_i^*)$ and $h^*(s_0, s_i^*)$ are equal to ∞ if no plan exists. Then, the reference solution set for the goal recognition task is

$$\Gamma^* = \{s_i^* \in \Gamma \mid \frac{h_\Omega^*(s_0, s_i^*)}{h^*(s_0, s_i^*)} \leq \frac{\text{cost}(\pi)}{\text{cost}(\pi^*)} \wedge h_\Omega^*(s_0, s_i^*) \neq \infty\}$$

In Figure 1 we show a goal recognition task with goal candidates $\Gamma = \{s_1^*, s_2^*\}$. Suppose that s_1^* is the reference goal. Then, $\Omega_1 = \langle \vec{o}_1 \rangle$ is an optimal sequence of observations because it is extracted from the optimal plan $\pi_1 = \langle o_1, o_2, o_3 \rangle$, $\Omega_2 = \langle \vec{o}_5, \vec{o}_7, \vec{o}_9 \rangle$ and $\Omega_3 = \langle \vec{o}_4, \dots, \vec{o}_{10} \rangle$ are sub-optimal sequences of observations because they are extracted from the sub-optimal plan $\pi_2 = \langle o_4, \dots, o_{10} \rangle$, and $\Omega_4 = \langle \vec{o}_4, \dots, \vec{o}_{10}, \vec{o}_{11} \rangle$ is a sub-optimal and noisy sequence of observations because it was extracted from π_2 and the observation of $\vec{o}_{11}$ was added. The reference solution set for goal recognition tasks with noisy observations is computed ignoring noisy observations in the sequence of observations. The reference solution set for any of these observation sequences with respective plans is $\Gamma_i^* = \{s_1^*\}$. For example,

$h_{\Omega_4}^*(s_0, s_1^*) = 7$, $h_{\Omega_4}^*(s_0, s_2^*) = 9$, $\text{cost}(\pi_2)/\text{cost}(\pi^*) = 7/3$ and thus $\Gamma_4^* = \{s_1^*\}$.

LP-Based Goal Recognition

We now develop an LP-based goal recognition method that expands the operator-counting framework with *observation-counting constraints*.

Observation-Counting Constraints We now introduce an IP/LP heuristic which expands the operator-counting framework with a set of *observation-counting constraints*. Definition 7 formally introduces the set of *observation-counting constraints* and the integer program that ensures that the solution computed satisfies all observation counts.

Definition 7 (Satisfying IP/LP heuristic). Let $\mathcal{Y}^\Omega$ be a set of non-negative integer variables with a variable $Y_{\vec{o}}$ for each operator $o \in \mathcal{O}$. Let $\text{occur}_\Omega(o)$ be the number of occurrences of operator o in Ω . Then, the satisfying integer program IP_Ω^C for a set of operator-counting constraints C , a set of observation-counting constraints, and sequence of observations Ω for state s is

$$\text{minimize } \sum_{o \in \mathcal{O}} \text{cost}(o) Y_o$$

subject to C ,

$$Y_{\vec{o}} \leq \text{occur}_\Omega(o) \quad \text{for all } o \in \mathcal{O} \quad (1)$$

$$Y_{\vec{o}} \leq Y_o \quad \text{for all } o \in \mathcal{O} \quad (2)$$

$$\sum_{Y_{\vec{o}} \in \mathcal{Y}^\Omega} Y_{\vec{o}} \geq |\Omega| \quad (3)$$

$$Y_o, Y_{\vec{o}} \in \mathbb{Z}_0^+.$$

The satisfying IP heuristic h_Ω^{IP} is the objective value of IP_Ω^C , and the satisfying LP heuristic h_Ω is the objective value of its linear relaxation. If the IP or LP is infeasible, the heuristic estimate is ∞ .

In the integer program, the set of constraints (1) limits the value of each $Y_{\vec{o}}$ by the number of occurrences of the operator o in Ω . Next, the set of constraints (2) binds the two sets $Y_{\vec{o}}$ and Y_o of variables. This set of constraints guarantees that Y_o acts as an upper bound for $Y_{\vec{o}}$. Thus, to increase the count of $Y_{\vec{o}}$ the integer program must first increase the count of Y_o which is minimized in the objective function and restricted by the set of operator-counting constraints C . Finally, constraint (3) ensures all observations are satisfied, since each $Y_{\vec{o}}$ is limited by the number of times o appears in Ω . While simpler models can compute the same objective value, we show how explicit information about the observations enables us to reason about noisy observations.

Note that $h_\Omega(s, s^*) \leq h_\Omega^*(s, s^*)$ for all states s of the planning task. First note that h is *admissible* (Pommerening et al. 2014) and that a complying plan π can always satisfy IP_Ω^C . Note that the only difference between IP_Ω^C and IP^C are the *observation-counting constraints*. These constraints only restrict the set of plans that can satisfy IP_Ω^C to the set of plans that satisfy all observations. If π is an optimal s -plan that complies with Ω ($h_\Omega^*(s, s^*) = \text{cost}(\pi)$), then there is a solution for $\text{IP}_\Omega^C/\text{LP}_\Omega^C$ where $Y_{\vec{o}} = Y_o = \text{occur}_\pi(o)$.

We use the h_Ω heuristic to estimate a lower bound on the cost of an optimal plan that satisfies all observations in Ω for each goal candidate in Γ . However, this information is insufficient to estimate the solution set because the goal candidate with the least h_Ω -value is not necessarily the most likely one. Consider a goal recognition task with two goal candidates. The first goal candidate has an optimal cost plan that can be extended with one operator to satisfy the single observation in Ω . The second goal candidate has an optimal cost plan that complies with the observation in Ω . In this example the plan for the first goal costs less than the plan for the second goal. In this example only the first goal candidate would be included in the solution set. However, we argue that the second goal is more likely to be part of the reference solution set since it is the only goal candidate with a complying optimal plan. Therefore, we normalize the values of h_Ω with estimates of the costs of the original optimal solution – without satisfying the observations. Like previous methods, the idea is to select the goals that have plans that satisfy all observations with the least additional cost. For this, we use the value of the original operator-counting heuristic h . Having h_Ω and h for each goal candidate we can compute the following solution set:

$$\delta_{\min} = \min_{s_i^* \in \Gamma : h_\Omega(s_0, s_i^*) < \infty} \{h_\Omega(s_0, s_i^*) - h(s_0, s_i^*)\} \quad (4)$$

$$\Gamma^{\text{LP}} = \{s_i^* \in \Gamma \mid h_\Omega(s_0, s_i^*) - h(s_0, s_i^*) = \delta_{\min}\} \quad (5)$$

Equation 4 computes the minimum difference $\delta_{\min}$ between the lower bound cost of an optimal plan that satisfies observations and the lower bound cost of an optimal plan (ignoring observations). The $\delta_{\min}$ value only considers goal candidates with bounded estimates for plans that satisfy observations ($h_\Omega(s_0, s^*) < \infty$). Equation 5 computes the solution set Γ^{LP} by selecting all goals with a difference between the estimates equal to $\delta_{\min}$. Note that Γ^{LP} is an approximated solution and not equal to the reference solution set Γ^* . In our running example, consider a goal recognition task with $\Omega = \langle \vec{o}_5, \vec{o}_7, \vec{o}_9 \rangle$. Then cost of $h_\Omega(s_0, s_1^*)$ and $h(s_0, s_1^*)$ are 7 and 3, and costs of $h_\Omega(s_0, s_2^*)$ and $h(s_0, s_2^*)$ are 9 and 3. Thus, $\delta_{\min}$ equals to 4, and we return $\Gamma^{\text{LP}} = \{s_1^*\}$.

Addressing Noisy Observations In most realistic settings, unreliable sensors may add noisy observations to the sequence of observations. Consider a goal recognition task in our running example with $\Omega = \langle \vec{o}_4, \dots, \vec{o}_{10}, \vec{o}_{11} \rangle$. Then, $h_\Omega(s_0, s_1^*) = 13$ and $h_\Omega(s_0, s_2^*) = 11$. In this situation we would have $\delta_{\min} = 8$, and $\Gamma^{\text{LP}} = \{s_2^*\}$. However, the observation $\vec{o}_{11}$ is unlikely to be part of any plan that generates the sequence of observations for either of the two goals. Evaluating precisely which observations are unlikely to be part of plans for a goal is a hard problem that requires solving a planning task multiple times, or, as Sohrabi, Riabov, and Udrea (2016) do, generating multiple plans. In spite of that, we can estimate the solution for this problem in polynomial time using the linear relaxation of an integer program. Specifically, we modify the integer program to try to automatically identify noisy observations given the unreliability of the sensors. The main modification is to replace constraint (3) in the integer program IP_Ω^C with constraint (6). We call the solution set using this heuristic Γ^ϵ .

$$\sum_{\gamma_{\sigma} \in \mathcal{Y}^{\Omega}} \gamma_{\sigma} \geq |\Omega| - \lfloor |\Omega| * \epsilon \rfloor \quad (6)$$

where ϵ is the *unreliability rating* of the sensor that represents the expected percentage of mistaken observations. This new constraint requires that at least $|\Omega| - \lfloor |\Omega| * \epsilon \rfloor$ observations be satisfied by the solution found. If $\epsilon = 0$, all observations must be satisfied. If $0 < \epsilon < 1$, some observations can be automatically ignored in order to minimize the objective value of h_{Ω} for each goal candidate. Consider our running example with $\Omega = \langle o_4, \dots, o_{10}, o_{11} \rangle$ and $\epsilon = 0.2$. Then, the integer program IP_{Ω}^C has to satisfy 7 observations. Then, $h_{\Omega}(s_0, s_1^*) = 7$ and $h_{\Omega}(s_0, s_2^*) = 9$. In this situation we would have $\delta_{\min} = 4$, and $\Gamma^{\epsilon} = \{s_1^*\}$.

Measuring Uncertainty The main idea of this approach is that if Ω has a lower percentage of observations we should be more careful in our decision. However, if the percentage is higher we can be more confident. A goal recognition task does not provide the percentage of observations in Ω . However, we can estimate this information using our heuristic h_{Ω} . Since h_{Ω} provides a lower bound on the cost of a plan that satisfies the observations the difference (if any) between h_{Ω} and $|\Omega|$ must be due to missing observations. Note that under 100% observability $h_{\Omega} = |\Omega|$, and with lower degrees of observability h_{Ω} may select operators that are not in Ω to satisfy the operator-counting constraints C . Thus, in lower degrees of observability the difference between h_{Ω} and $|\Omega|$ is likely to increase. This information allows us to estimate the degree of observability as follows:

$$\mu = 1 + \frac{\max_{s_i^* \in \Gamma^{\text{LP}}} \{h_{\Omega}(s_0, s_i^*)\} - |\Omega|}{\max_{s_i^* \in \Gamma^{\text{LP}}} \{h_{\Omega}(s_0, s_i^*)\}} \quad (7)$$

where μ is the *uncertainty ratio*. This value is computed by first selecting the goal candidates using Equation 5, and then selecting the goal candidate in Γ^{LP} with maximum h_{Ω} . Having μ we can compute the new solution set Γ^{μ} that considers uncertainty:

$$\Gamma^{\mu} = \{s_i^* \in \Gamma \mid h_{\Omega}(s_0, s_i^*) - h(s_0, s_i^*) \leq \delta_{\min} * \mu\} \quad (8)$$

Consider our running example with $\Omega = \langle \vec{o}_6 \rangle$. Then, $h_{\Omega}(s_0, s_1^*) = 7$ and $h_{\Omega}(s_0, s_2^*) = 9$. In this situation we would have $\delta_{\min} = 4$, and $\Gamma^{\text{LP}} = \{s_1^*\}$. However, we would argue that having only one observation is insufficient to make a precise decision. Using uncertainty we would have $\mu = 1 + 6/7$ and $\Gamma^{\mu} = \{s_1^*, s_2^*\}$.

Experimental Results

We conducted extensive empirical experiments to show the effectiveness of our methods in three ways. First, we evaluate how each source of operator-counting constraints impacts the quality of our solutions. Second, we assess the performance of our methods that explicitly address low observability and noise. Finally, we compare our approach with previous ones. We ran all experiments with Ubuntu running over an Intel Core i7 930 CPU (2.80 GHz) with a memory limit of 1 GB, all methods solved each goal recognition task under a time limit of five seconds. Our implementation uses

Fast Downward version 19.06 (Helmert 2006), a Python pre-possessing layer, and the CPLEX 12.10 LP solver.¹

Benchmark Domains and Data sets We create a new benchmark by adapting the one introduced by Pereira, Oren, and Meneguzzi (2017) to use the *agreement ratio* evaluation metric from Ramírez and Geffner (2009). For each domain we create three planning tasks each (except for IPC-GRID, in which we create four) with four reference goals each that we use to compute the plans from which we extract the sequence of observations. We compute optimal and sub-optimal plans for each pair of planning task and reference goal creating two data sets. To compute sub-optimal plans we use weighted A^* with $w = 2$ (Pohl 1970).

Following previous work, we experiment with five different levels of observability: 10%, 30%, 50%, 70% and 100%. We only generate one sequence of observations for 100% of observability, and three different random observation sequences from the same plan for other percentages, generating 208 goal recognition tasks in total for IPC-GRID and 156 for each of the other domains in each data set (optimal and sub-optimal). For each data set we also create a corresponding noisy data set by adding $\lceil |\Omega| * 0.2 \rceil$ randomly generated observations in each sequence of observations— i.e. the fault rate of the sensor is 20%. Three different noisy sequences are generated for each original sequence. For each goal recognition task we add at least five randomly generated candidate goal conditions. In total we have 8,288 goal recognition tasks divided in four data sets. In order to create the new benchmarks, we compute the reference solution set Γ^* for each goal recognition task for optimal and sub-optimal data sets. Thus, for each goal candidate of each base task we solve a planning task twice.

We evaluate the methods using three metrics: *agreement ratio*, *accuracy* and *spread*. The *agreement ratio* is defined as the intersection over union $|\Gamma^* \cap \Gamma| / |\Gamma^* \cup \Gamma|$ of the reference solution set Γ^* against the solution Γ provided by the method. The *accuracy* is 1 if the solution set chosen by the evaluated method contains the reference goal and 0 otherwise. Note that we use a slight modified of accuracy in other to compare to (Pereira, Oren, and Meneguzzi 2017). The *spread* is the size of the solution set chosen by the evaluated method. Table 1 summarizes the information about the data sets. The domains we use are BLOCKS WORLD, DEPOTS, DRIVERLOG, DWR, IPC GRID, FERRY, LOGISTICS, MICONIC, ROVERS, SATELLITE, SOKOBAN and ZENO TRAVEL. Due to space restrictions, we summarize results for all domains as averages in OTHER, except for BLOCKS WORLD, IPC GRID and SOKOBAN. For each domain row, $|\Gamma|$ is the average number of candidate goals. Columns $|\Omega|$ and $|\Gamma^*|$ show the average sizes of the observations and the reference solution set, respectively. The average size of the plan with 100% of observability indicates the size of the plan computed for the reference goal. As expected, the average sizes $|\Omega|$ and $|\Gamma^*|$ are larger for the sub-optimal data set than for the optimal data set.

¹Source-code and benchmark are available at: <https://bit.ly/lp-goal-recognition>

		Optimal				Sub-Optimal			
#	%	$ \Gamma $	$ \Omega $	$ \Gamma^* $	$ \Omega $	$ \Gamma^* $	$ \Omega $	$ \Gamma^* $	
BLOCKS	10		1.25	8.0	1.42	7.61			
	30		3.08	3.97	3.83	3.58			
	50	20.33	4.42	2.5	5.92	3.19			
	70		6.67	1.94	8.5	2.53			
	100		8.83	1.83	11.83	2.25			
IPC-GRID	10		1.63	2.71	2.06	1.58			
	30		4.0	1.21	5.56	1.4			
	50	7.5	6.19	1.13	8.88	1.35			
	70		8.69	1.04	12.56	1.31			
	100		11.88	1.0	17.25	1.5			
SOKOBAN	10		2.33	2.11	3.33	1.83			
	30		6.5	1.25	8.67	1.28			
	50	8.67	10.33	1.22	13.75	1.33			
	70		14.67	1.03	19.33	1.36			
	100		20.17	1.0	27.0	1.33			
OTHER	10		1.85	3.01	2.46	2.32			
	30		4.69	1.61	6.37	1.45			
	50	6.89	7.52	1.21	10.04	1.21			
	70		10.61	1.1	14.13	1.15			
	100		14.51	1.06	19.55	1.08			

Table 1: Key properties of each experimental domain.

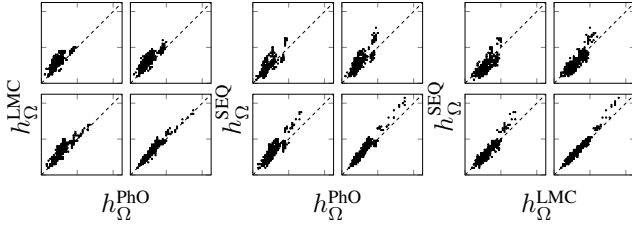


Figure 2: Heuristic values for h_Ω .

Evaluating the Constraints We measure how the sources of operator-counting constraints impact the quality of the solutions, and if more informed heuristics h_Ω improve the solution of goal recognition tasks. The constraint sources are: state equation h_Ω^{SEQ} (Bonet 2013), landmarks h_Ω^{LMC} (Bonet and van den Briel 2014), and the post-hoc optimization h_Ω^{PhO} (Pommerening, Röger, and Helmert 2013).

Figure 2 shows the value of h_Ω for each source of constraints. Each point is the h_Ω -value for a goal recognition task with its reference goal in the optimal data set. There are four figures in each group, one for each degree of observability (10%, 30%, 50% and 70%). They show that, in general, h_Ω^{SEQ} and h_Ω^{LMC} are more informed than h_Ω^{PhO} , and that h_Ω^{SEQ} and h_Ω^{LMC} are comparable. Also, as expected, the difference in the values decreases as observability increases. On average h_Ω^{SEQ} and h_Ω^{LMC} are more informed than h_Ω^{PhO} on 61.49% and 72.58% of the goal recognition tasks respectively. h_Ω^{SEQ} is more informed than h_Ω^{LMC} on 31.19% of the tasks, and h_Ω^{LMC} is more informed than h_Ω^{SEQ} on 42.96% of the tasks.

Table 2 shows the results of the *agreement* ratio for each source of operator-counting constraints solving goal recognition tasks in the optimal and sub-optimal data sets. The solution set Γ^{LP} is computed using h_Ω , and when two or

		Optimal								Sub-Optimal							
#	%	S	L	P	S	L	P	S	P	S	L	P	S	L	P	S	P
BLOCKS	10	0.45	0.42	0.44	0.45	0.41	0.44	0.44	0.41	0.39	0.44	0.39	0.41	0.5	0.44	0.49	
	30	0.43	0.33	0.43	0.43	0.47	0.47	0.5	0.44	0.41	0.5	0.44	0.5	0.44	0.49		
	50	0.55	0.46	0.44	0.55	0.58	0.59	0.64	0.45	0.55	0.64	0.45	0.55	0.64	0.45	0.55	
	70	0.75	0.54	0.58	0.75	0.81	0.85	0.74	0.52	0.58	0.74	0.52	0.58	0.74	0.52	0.58	
	100	0.82	0.58	0.62	0.82	0.88	0.92	0.82	0.58	0.62	0.82	0.58	0.62	0.82	0.58	0.62	
IPC-GRID	10	0.65	0.92	0.4	0.87	0.92	0.68	0.6	0.86	0.25	0.76	0.86	0.63	0.69	0.88	0.23	0.82
	30	0.73	0.97	0.25	0.93	0.97	0.78	0.81	0.89	0.29	0.84	0.89	0.87	0.87	0.91	0.08	0.89
	50	0.83	0.97	0.27	0.96	0.97	0.9	0.87	0.91	0.08	0.89	0.91	0.89	0.89	0.94	0.04	0.94
	70	0.9	0.97	0.3	0.97	0.97	0.95	0.94	0.94	0.05	0.94	0.94	0.94	0.94	0.94	0.94	0.94
	100	1.0	1.0	0.23	1.0	1.0	1.0	0.94	0.94	0.05	0.94	0.94	0.94	0.94	0.94	0.94	0.94
SOKOBAN	10	0.38	0.38	0.24	0.39	0.34	0.31	0.38	0.3	0.24	0.52	0.25	0.36	0.72	0.43	0.14	0.77
	30	0.59	0.41	0.14	0.75	0.38	0.59	0.77	0.51	0.17	0.79	0.41	0.79	0.81	0.89	0.29	0.84
	50	0.82	0.53	0.21	0.92	0.49	0.82	0.85	0.58	0.17	0.85	0.58	0.17	0.85	0.58	0.17	0.85
	70	0.93	0.73	0.21	0.99	0.62	0.93	0.88	0.73	0.22	0.88	0.73	0.22	0.88	0.73	0.22	0.88
	100	0.96	0.85	0.23	1.0	0.81	0.96	0.63	0.63	0.55	0.72	0.63	0.64	0.78	0.69	0.69	0.78
OTHER	10	0.71	0.73	0.63	0.78	0.72	0.69	0.67	0.7	0.54	0.78	0.69	0.69	0.8	0.78	0.61	0.87
	30	0.71	0.71	0.54	0.82	0.7	0.73	0.8	0.78	0.61	0.87	0.78	0.83	0.9	0.87	0.64	0.94
	50	0.81	0.78	0.58	0.88	0.77	0.82	0.9	0.87	0.63	0.96	0.87	0.92	0.95	0.93	0.66	0.97
	70	0.91	0.87	0.63	0.96	0.87	0.92	0.95	0.93	0.66	0.97	0.95	0.95	0.97	0.95	0.95	0.97
	100	0.96	0.94	0.66	0.98	0.95	0.95	0.76	0.74	0.52	0.82	0.75	0.78	0.82	0.75	0.78	0.82
AVG		0.79	0.77	0.54	0.86	0.78	0.8	0.76	0.74	0.52	0.82	0.75	0.78	0.82	0.75	0.78	0.82

Table 2: Agreement ratio for constraint sets state equation h_Ω^{SEQ} (S), landmarks h_Ω^{LMC} (L), and post-hoc h_Ω^{PhO} (P).

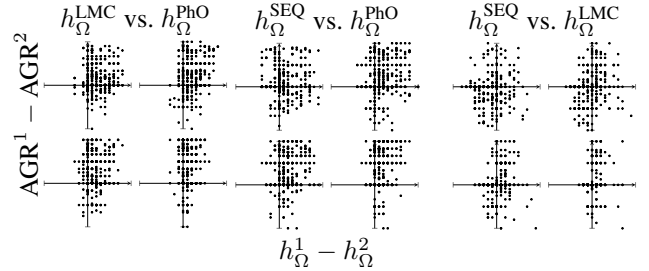


Figure 3: Relation between h_Ω -value and agreement ratio.

more sources of operator-counting constraints are used, they are all combined into a single integer program IP_Ω^C . The first group of columns shows the results for each source of constraints used individually, and the second combined in pairs. When the constraints are used individually h_Ω^{SEQ} and h_Ω^{LMC} achieve the best results for different domains. For example, h_Ω^{SEQ} is the best for BLOCKS while h_Ω^{LMC} is the best for IPC-GRID. When pairs of constraints are combined the results improve and again the pair formed by h_Ω^{LMC} and h_Ω^{SEQ} provides best results. Results using all constraints are similar to using the pair h_Ω^{LMC} and h_Ω^{SEQ} (as presented next in Table 3). There are two key conclusions of these results. First, the agreement increases with the degree of observability, but even with 100% it is still hard to obtain perfect agreement. Second, the agreement degrades in the sub-optimal data set, but our method maintains an average of 0.82.

Figure 3 shows the relation between h_Ω -value and agreement for goal recognition tasks. Again, there are four figures in each group, one for each degree of observability. Each point on the x -axis shows the difference between the

		Optimal																	
		Γ^{LP}			Γ^{μ}			RG			POM			POM-10%			POM-30%		
#	%	AGR	ACC	SPR	AGR	ACC	SPR	AGR	ACC	SPR	AGR	ACC	SPR	AGR	ACC	SPR	AGR	ACC	SPR
BLOCKS	10	0.44	0.86	7.53	0.44	0.86	7.56	0.47	0.92	9.83	0.06	0.17	1.44	0.13	0.47	4.06	0.38	1.0	18.14
	30	0.46	0.78	2.5	0.44	0.86	4.67	0.45	0.92	5.56	0.21	0.39	1.17	0.3	0.75	2.94	0.24	1.0	15.25
	50	0.59	0.89	3.03	0.52	0.89	3.86	0.62	0.97	3.69	0.33	0.58	1.25	0.37	0.81	3.08	0.25	0.97	12.17
	70	0.85	0.97	1.83	0.76	0.97	2.42	0.81	1.0	2.22	0.51	0.72	1.14	0.45	0.94	2.19	0.25	1.0	9.22
	100	0.92	1.0	1.67	0.92	1.0	1.67	0.9	1.0	2.08	0.59	1.0	1.67	0.55	1.0	1.92	0.31	1.0	6.42
IPC-GRID	10	0.87	0.94	2.67	0.88	0.96	2.69	0.91	1.0	3.23	0.47	0.75	2.35	0.55	0.98	4.38	0.49	1.0	6.25
	30	0.93	0.96	1.15	0.94	0.98	1.17	0.99	1.0	1.25	0.85	0.98	1.52	0.81	1.0	1.96	0.64	1.0	3.17
	50	0.96	0.98	1.08	0.96	0.98	1.08	1.0	1.0	1.13	0.86	1.0	1.44	0.86	1.0	1.56	0.77	1.0	2.15
	70	0.97	0.98	1.06	0.97	0.98	1.06	1.0	1.0	1.04	0.97	0.98	1.02	0.97	0.98	1.02	0.93	0.98	1.15
	100	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
SOKOBAN	10	0.39	0.53	2.08	0.38	0.61	2.94	0.4	0.81	4.86	0.28	0.53	2.14	0.32	0.89	4.39	0.26	0.97	7.0
	30	0.75	0.81	1.25	0.64	0.92	2.06	0.56	0.86	2.53	0.57	0.69	1.22	0.48	0.75	1.89	0.23	0.94	5.17
	50	0.92	1.0	1.19	0.83	1.0	1.39	0.61	0.86	2.14	0.61	0.69	1.42	0.55	0.81	2.14	0.28	1.0	5.08
	70	0.99	1.0	1.0	0.94	1.0	1.08	0.64	0.83	1.53	0.85	0.92	1.17	0.81	0.94	1.39	0.36	1.0	3.64
	100	1.0	1.0	1.0	1.0	1.0	1.0	0.67	0.75	1.17	1.0	1.0	1.0	1.0	1.0	1.0	0.42	1.0	2.75
OTHER	10	0.78	0.89	3.27	0.78	0.9	3.38	0.74	0.93	3.88	0.4	0.47	1.67	0.53	0.77	3.68	0.46	0.99	6.43
	30	0.81	0.91	1.77	0.79	0.94	2.04	0.69	0.95	2.67	0.63	0.73	1.38	0.57	0.89	2.54	0.3	0.99	5.59
	50	0.88	0.95	1.32	0.84	0.97	1.63	0.77	0.95	1.86	0.77	0.85	1.16	0.7	0.93	1.84	0.29	0.98	4.8
	70	0.95	0.99	1.15	0.94	0.99	1.23	0.82	0.96	1.49	0.88	0.94	1.13	0.78	0.99	1.55	0.35	1.0	3.92
	100	0.98	1.0	1.06	0.98	1.0	1.06	0.9	0.97	1.24	0.95	1.0	1.08	0.87	1.0	1.29	0.45	1.0	3.01
AVG		0.86	0.94	1.79	0.84	0.95	1.99	0.77	0.95	2.39	0.7	0.79	1.31	0.67	0.91	2.22	0.39	0.99	5.2
Sub-Optimal																			
		Γ^{LP}			Γ^{μ}			RG			POM			POM-10%			POM-30%		
#	%	AGR	ACC	SPR	AGR	ACC	SPR	AGR	ACC	SPR	AGR	ACC	SPR	AGR	ACC	SPR	AGR	ACC	SPR
BLOCKS	10	0.41	0.86	6.86	0.42	0.89	7.42	0.46	0.97	10.61	0.06	0.19	1.19	0.19	0.58	4.44	0.34	1.0	17.53
	30	0.49	0.78	3.17	0.35	0.86	6.92	0.54	1.0	4.86	0.28	0.56	1.17	0.32	0.89	3.36	0.26	1.0	13.47
	50	0.55	0.86	3.08	0.42	0.94	5.61	0.62	0.97	2.72	0.39	0.72	1.08	0.36	0.81	2.17	0.27	1.0	9.89
	70	0.71	0.92	2.06	0.56	0.94	3.06	0.68	1.0	2.44	0.51	0.94	1.33	0.44	1.0	2.22	0.26	1.0	8.61
	100	0.84	1.0	1.67	0.84	1.0	1.67	0.8	1.0	2.08	0.51	1.0	1.67	0.48	1.0	1.92	0.28	1.0	6.42
IPC-GRID	10	0.77	0.92	1.81	0.75	0.98	2.4	0.81	1.0	2.73	0.64	0.88	2.23	0.59	0.98	3.44	0.42	1.0	5.6
	30	0.82	0.94	1.13	0.77	0.98	1.6	0.9	1.0	1.27	0.81	0.96	1.35	0.8	0.96	1.52	0.67	0.96	2.23
	50	0.84	0.94	1.13	0.84	1.0	1.56	0.92	1.0	1.1	0.87	1.0	1.08	0.86	1.0	1.13	0.86	1.0	1.21
	70	0.89	1.0	1.1	0.85	1.0	1.23	0.93	1.0	1.02	0.92	1.0	1.0	0.92	1.0	1.0	0.92	1.0	1.0
	100	0.94	1.0	1.0	0.94	1.0	1.0	0.94	1.0	1.0	0.94	1.0	1.0	0.94	1.0	1.0	0.94	1.0	1.0
SOKOBAN	10	0.52	0.61	1.78	0.44	0.72	3.17	0.29	0.64	4.56	0.35	0.64	2.47	0.38	0.92	4.08	0.24	1.0	6.86
	30	0.77	0.83	1.08	0.62	0.97	2.67	0.43	0.75	2.92	0.56	0.75	1.72	0.51	0.86	2.64	0.24	0.97	5.5
	50	0.79	0.92	1.17	0.66	1.0	2.58	0.53	0.72	1.83	0.58	0.75	1.39	0.53	0.86	2.31	0.25	0.97	5.14
	70	0.8	0.97	1.03	0.85	1.0	1.39	0.54	0.61	1.28	0.63	0.86	1.25	0.59	0.92	1.75	0.3	1.0	4.11
	100	0.83	1.0	1.0	0.83	1.0	1.0	0.58	0.58	1.33	0.83	1.0	1.0	0.83	1.0	1.0	0.38	1.0	2.75
OTHER	10	0.72	0.89	2.84	0.7	0.92	3.29	0.63	0.94	3.81	0.45	0.56	1.69	0.48	0.82	3.63	0.37	0.99	6.35
	30	0.77	0.9	1.74	0.63	0.95	2.89	0.68	0.93	2.35	0.65	0.78	1.32	0.55	0.91	2.48	0.28	0.99	5.47
	50	0.87	0.95	1.33	0.72	0.99	2.23	0.76	0.96	1.85	0.79	0.9	1.22	0.69	0.97	1.9	0.3	1.0	4.59
	70	0.94	0.98	1.16	0.83	0.99	1.63	0.84	0.96	1.45	0.87	0.95	1.1	0.79	0.98	1.52	0.37	1.0	3.81
	100	0.97	1.0	1.06	0.97	1.0	1.06	0.89	0.97	1.24	0.95	1.0	1.08	0.86	1.0	1.29	0.46	1.0	3.01
AVG		0.82	0.93	1.71	0.75	0.97	2.39	0.74	0.93	2.3	0.7	0.83	1.31	0.65	0.93	2.19	0.38	1.0	5.01

Table 3: Agreement ratio (AGR), accuracy (ACC) and spread (SPR) for each method on optimal and sub-optimal data sets.

agreement ratio for the solution generated using h_Ω for each goal recognition task in the optimal data set (-1 to 1). The y -axis shows the difference of h_Ω -values for the same task with its goal reference (-10 to 10). For example, in group h_Ω^{LMC} vs. h_Ω^{PhO} the points are clustered on right upper quadrant which shows that in general when AGR^1 is higher than AGR^2 , h_Ω^1 is also higher than h_Ω^2 . We see the same trend in group h_Ω^{SEQ} vs. h_Ω^{PhO} . We highlight a different situation in the group h_Ω^{SEQ} vs. h_Ω^{LMC} since the points are clustered on the upper right and bottom left quadrants. This shows that higher h_Ω -values tend to produce higher agreement ratio. Thus, these results provide evidence that more informed heuristics improve the solution of goal recognition tasks.

Evaluating Previous Methods and Uncertainty Table 3 compares our Γ^{LP} and Γ^μ methods to two other polynomial time approaches from the literature, reporting agreement ratio, accuracy and spread for optimal and sub-optimal data sets with degrees of observability with 10%, 30%, 50%, 70% and 100% of the observations. Our methods use the three sources of constraints h^{SEQ} , h^{LMC} , and h^{PhO} . RG (Ramírez and Geffner 2009) computes a relaxed plan efficiently and returns as the goal set the goal candidates with relaxed plans that satisfy the largest number of observations. POM (Pereira, Oren, and Meneguzzi 2017) performs the recognition task by computing landmarks and returns as the goal set the goal candidates that have the highest number of landmarks satisfied by the observation. We use their *goal completion* heuristic for its better results. We report the results of POM-10% and POM-30%, which return larger goal sets, including those within a 10% and 30% threshold of the goals with the highest number of landmarks satisfied.

On both data sets our approach Γ^{LP} has the highest agreement ratio on average and is the best in almost all domains and degrees of observability. An exception is the domain IPC-GRID where RG has in general better results. We note that in hard domains like SOKOBAN, our methods have much higher agreements ratios than other approaches. For example, on the optimal data set for this domain, Γ^{LP} has average agreement ratio of 0.81 while the next best approach RG has average agreement ratio of 0.76.

Table 3 also shows accuracy and spread for all methods. It shows that many methods can achieve high accuracy while yielding a high spread, thus degrading the agreement ratio. For example, while POM-30% has a perfect accuracy on almost all domains on the sub-optimal data set, its spread is the highest. The BLOCKS domain has on average 20.33 goal candidates, and for POM-30% to achieve a competitive accuracy on 10% of observability it returns almost all goals with a spread of 17.53. By contrast, our Γ^μ method increases the accuracy without increasing the spread excessively by measuring uncertainty. This happens especially in the low observability scenarios it was designed to address. Take for example the results of SOKOBAN on sub-optimal data set, in which Γ^μ shows a substantially higher accuracy without a corresponding increase in spread, unlike other methods. Our idea to measure the uncertainty is general since it does not require linear programming heuristics and could be applied to RG and POM to improve their results.

#	%	Optimal				Sub-Optimal			
		Γ^{LP}	Γ^ϵ	RG	POM	Γ^{LP}	Γ^ϵ	RG	POM
BLOCKS	10	0.32	0.32	0.31	0.06	0.38	0.38	0.42	0.05
	30	0.37	0.37	0.39	0.13	0.36	0.37	0.49	0.22
	50	0.64	0.64	0.6	0.37	0.53	0.53	0.55	0.28
	70	0.79	0.81	0.77	0.47	0.67	0.67	0.63	0.38
	100	0.88	0.88	0.89	0.57	0.78	0.82	0.74	0.51
IPC-GRID	10	0.57	0.57	0.16	0.38	0.62	0.62	0.12	0.54
	30	0.85	0.85	0.28	0.71	0.68	0.68	0.08	0.72
	50	0.89	0.89	0.07	0.81	0.84	0.84	0.04	0.85
	70	0.95	0.95	0.15	0.93	0.89	0.89	0.02	0.9
	100	1.0	1.0	0.08	0.99	0.94	0.94	0.04	0.92
SOKOBAN	10	0.27	0.27	0.1	0.24	0.31	0.32	0.13	0.25
	30	0.56	0.7	0.2	0.34	0.48	0.56	0.12	0.29
	50	0.61	0.82	0.2	0.57	0.5	0.73	0.01	0.46
	70	0.6	0.97	0.08	0.84	0.54	0.8	0.06	0.58
	100	0.66	1.0	0.04	0.96	0.35	0.85	0.04	0.77
OTHER	10	0.44	0.44	0.27	0.27	0.42	0.42	0.31	0.31
	30	0.57	0.58	0.32	0.51	0.6	0.63	0.36	0.53
	50	0.77	0.78	0.37	0.65	0.78	0.78	0.41	0.71
	70	0.88	0.89	0.46	0.8	0.86	0.88	0.41	0.79
	100	0.95	0.96	0.45	0.9	0.91	0.96	0.4	0.87
AVG		0.71	0.73	0.35	0.61	0.69	0.72	0.34	0.61
ACC		0.89	0.9	0.5	0.73	0.87	0.89	0.48	0.76
SPR		1.91	1.78	1.38	1.3	1.93	1.72	1.34	1.31

Table 4: Agreement ratio, and average accuracy (ACC) and spread (SPR) results on data sets with noisy observations.

Noisy Observations Table 4 compares agreement ratio of our Γ^{LP} and Γ^ϵ methods with RG and POM on noisy data sets. The last two rows show the average accuracy and spread over all domains. Again, our methods use the three sources of constraints h^{SEQ} , h^{LMC} , and h^{PhO} . Here most methods degrade with noisy observations, reducing their agreement ratio. Γ^ϵ , which addresses noisy observations explicitly, has on average the highest agreement ratio and accuracy on both data sets. For example, on the SOKOBAN domain some noisy observations might be impossible to satisfy because they lead to unsolvable states on all plans. In this situation Γ^ϵ substantially improves the agreement ratio.

Discussion

In this paper we developed a novel class of goal recognition methods based on linear programming models. These methods include an uncertainty measurement that increases the accuracy on low observability scenarios, as well as an efficient and automatic method to address noisy observations. We adapt and provide a benchmark to compare methods using the agreement ratio, which allows us to evaluate our methods in a number of different ways. First, we evaluate how different sources of constraints impact the quality of our solutions. Second, we assess how our additional constraints and uncertainty measurement affect performance under noise and low observability, respectively. Third, we compare our methods to previous ones, showing that ours are, in general, superior.

Acknowledgments

Felipe Meneguzzi acknowledges support from CNPq with projects 407058/2018-4 (Universal) and 302773/2019-3 (PQ Fellowship). André G. Pereira acknowledges support from FAPERGS with project 17/2551-0000867-7. This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001. Ramon Fraga Pereira acknowledges support from the ERC Advanced Grant WhiteMech (No. 834228) and the EU ICT-48 2020 project TAILOR (No. 952215).

References

- Bonet, B. 2013. An Admissible Heuristic for SAS⁺ Planning Obtained from the State Equation. In *International Joint Conference on Artificial Intelligence*, 2268–2274.
- Bonet, B.; and van den Briel, M. 2014. Flow-based heuristics for optimal planning: Landmarks and merges. In *International Conference on Automated Planning and Scheduling*, 47–55.
- E-Martín, Y.; R.-Moreno, M. D.; and Smith, D. E. 2015. A Fast Goal Recognition Technique Based on Interaction Estimates. In *International Joint Conference on Artificial Intelligence*.
- Harman, H.; and Simoens, P. 2020. Action Graphs for Goal Recognition Problems with Inaccurate Initial States. In *AAAI Conference on Artificial Intelligence*.
- Helmert, M. 2006. The Fast Downward planning system. *Journal of Artificial Intelligence Research* 26: 191–246.
- Pereira, R. F.; Oren, N.; and Meneguzzi, F. 2017. Landmark-based heuristics for goal recognition. In *AAAI Conference on Artificial Intelligence*.
- Pohl, I. 1970. Heuristic search viewed as path finding in a graph. *Artificial intelligence* 1(3-4): 193–204.
- Pommerening, F.; Röger, G.; and Helmert, M. 2013. Getting the most out of pattern databases for classical planning. In *International Joint Conference on Artificial Intelligence*, 2357–2364.
- Pommerening, F.; Röger, G.; Helmert, M.; and Bonet, B. 2014. LP-based heuristics for cost-optimal planning. In *International Conference on Automated Planning and Scheduling*, 226–234.
- Ramírez, M.; and Geffner, H. 2009. Plan recognition as planning. In *International Joint Conference on Artificial intelligence*, 1778–1783.
- Ramírez, M.; and Geffner, H. 2010. Probabilistic plan recognition using off-the-shelf classical planners. In *AAAI Conference on Artificial Intelligence*, 1121–1126.
- Sohrabi, S.; Riabov, A. V.; and Udrea, O. 2016. Plan Recognition as Planning Revisited. In *International Joint Conference on Artificial intelligence*, 3258–3264.