Generalized Potential Heuristics for Classical Planning

Guillem Francès, Augusto B. Corrêa, Cedric Geissmann, Florian Pommerening

University of Basel, Switzerland

February 7, 2020

Originally presented at IJCAI'19

Generalized Potential Heuristics

In this Work

- Context: classical planning using heuristic search
- Some domains can be solved in linear time
- Hill-climbing + heuristic leading direct to the goal
- We want to learn these heuristics automatically

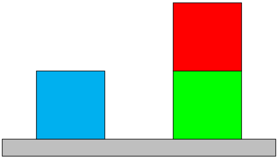
Generalized Potential Heuristics

Definition (Generalized Potential Heuristic)

Linear combination of features well-defined over all tasks:

$$h(s) = \sum_{f \in \mathcal{F}} w(f) \cdot f(s)$$

Description Logics for Planning



Primitive Concepts & Roles

- $\mathit{ontable} = \{{\color{cyan}\blacksquare}, {\color{green}\blacksquare}\}$
- $\mathit{on} = \{({\color{red}\blacksquare}, {\color{green}\blacksquare})\}$
- $\mathit{holding} = \emptyset$
- $\mathit{clear} = \{{\color{cyan}\blacksquare}, {\color{red}\blacksquare}\}$
- $\mathit{clear}_G = \{{\color{green}\blacksquare}\}$

Description Logic

Description Logic $\mathcal{SOI}$ with Role Value Maps

Complex concepts

- $\bot$, $\top$
- $\neg C$
- $C_1 \sqcup C_2$, $C_1 \sqcap C_2$
- $\forall R.C$, $\exists R.C$
- $R_1 = R_2$
- $\{a_1, \ldots, a_n\}$

Complex roles

- R^{-1}
- R^{+}
- $R_1 \circ R_2$

Complex Concepts & Roles

- "Set of blocks above some block that needs to be cleared"

$$\exists on^{+}.clear_G = \{\blacksquare\}$$

Example: Clearing a Block

Generalized Potential Heuristic for Blocksworld

→ Blocksworld tasks where the goal is to clear a set of blocks

$$h(s) = 2 \cdot |C_1| + |C_2|$$

- $C_1 \equiv \exists on^+.clear_G$:
  "Set of blocks above some block that needs to be cleared"
- $C_2 \equiv holding$:
  "Set of blocks being held"

Existence

- We prove that generalized heuristics leading directly to a goal state exist for a number of standard domains:
  - Blocksworld
  - Gripper
  - Spanner
  - Logistics
  - ...
- Greedy search solves any task in time $O(|\text{Objects}|)$

Existence

- We prove that generalized heuristics leading directly to a goal state exist for a number of standard domains:
  - Blocksworld
  - Gripper
  - Spanner
  - Logistics
  - ...
- Greedy search solves any task in time $O(|\text{Objects}|)$

Can we obtain these heuristics automatically?

Learning the Heuristic

Learning the Heuristic

Overview of our inductive approach:

1. Fully expand small tasks to generate training set $\mathcal{S}$.

Learning the Heuristic

Overview of our inductive approach:

1. Fully expand small tasks to generate training set $\mathcal{S}$.
2. Generate set of generalized features $\mathcal{F}$ with all features under a certain syntactic complexity.

Learning the Heuristic

Overview of our inductive approach:

1. Fully expand small tasks to generate training set $\mathcal{S}$.
2. Generate set of generalized features $\mathcal{F}$ with all features under a certain syntactic complexity.
3. Compute simplest potential heuristic on $\mathcal{F}$ leading states from $\mathcal{S}$ directly to the goal.
   - If no such h exists, augment $\mathcal{F}$.
   - If it does exist, test h on unseen tasks.

Computing the Weights

Mixed Integer Linear Program

$$\min_{w} \sum_{f \in \mathcal{F}} [w_f \neq 0] \mathcal{K}(f) \quad \text{subject to}$$

$$\bigvee_{s' \in succ(s)} h(s') + 1 \leq h(s) \quad \text{for alive states } s$$

$$h(s') \geq h(s) \quad \text{for transitions } (s, s') \text{ where } s \text{ is alive and } s' \text{ is unsolvable}$$

Results

- Learn: Generalized heuristics on standard domains
  - Gripper, Miconic, Spanner, VisitAll.
- Prove: Heuristics generalize all possible tasks.
  - Except VisitAll: No linear solution possible
- Solve: Steepest-ascent hill-climbing in linear time.
- In some domains, such heuristics exist but we cannot scale.

Conclusion

Contributions

- Generalized descending and dead-end avoiding heuristics exist for several planning domains.
- Heuristics learned can be intepreted.
- Solve any task in linear time.
- We can learn them automatically from a suitable logical model and small tasks.
  - Heuristic refinement procedure in the paper

Bonus Slides

Semantics – Examples

Concepts:

$$(\exists R.C)^{\mathcal{M}} = \{a \mid \exists b : (a, b) \in R^{\mathcal{M}} \wedge b \in C^{\mathcal{M}}\},$$
$$(R = R')^{\mathcal{M}} = \{a \mid \forall b : (a, b) \in R^{\mathcal{M}} \leftrightarrow (a, b) \in R'^{\mathcal{M}}\}.$$

Roles:

$$(R^{-1})^{\mathcal{M}} = \{(b, a) \mid (a, b) \in R^{\mathcal{M}}\},$$
$$(R \circ R')^{\mathcal{M}} = \{(a, c) \mid \exists b : (a, b) \in R^{\mathcal{M}} \wedge (b, c) \in R'^{\mathcal{M}}\}$$

Example: unrestricted Blocksworld tasks

$$h_{\text{bw}}(s) = -4|C_6| - |\mathit{holding}| - 2|\mathit{ontable}| - 2|C_7|,$$

- C_1 : $\mathit{ontable}_G \sqcap \mathit{ontable}$
  Blocks that are correctly placed on the table
- C_2: $(\exists \mathit{on}_G.\top) \sqcap (\mathit{on} = \mathit{on}_G)$
  Blocks that are placed on their target block
- C_3: $\neg(\mathit{ontable}_G \sqcup \exists \mathit{on}_G.\top)$
  Blocks that are not mentioned in the goal
- C_4: $C_1 \sqcup C_2 \sqcup C_3$
  Blocks where block (or table) below is consistent with the goal
- C_5: $\forall \mathit{on}_G^{-1}.(\mathit{on} = \mathit{on}_G)$
  Blocks where the block above is consistent with the goal
- C_6: $C_4 \sqcap \forall \mathit{on}^+.(C_4 \sqcap C_5)$
  Blocks that are well-placed.
- C_7: $\mathit{holding} \sqcap \exists \mathit{on}_G.(\mathit{clear} \sqcap C_6)$
  Blocks held while their target block is clear and well-placed.

	G	M	S	V
# of training instances	8	12	11	9
# of iterations	2.0	2.7	1.0	1.7
$\lvert\mathcal{F}\rvert$	469	2105	904	330
# of MIP variables	2017	7273	3381	1039
# of MIP constraints	2238	7331	3370	1190
Complexity of h	8 (18)	6 (14)	8 (20)	5 (8)
# of features in h	5	4	5	3
Total time	8h	32m	178s	87s
Total MIP time	7.4h	26m	6.8s	2.1s