

Planning and Optimization

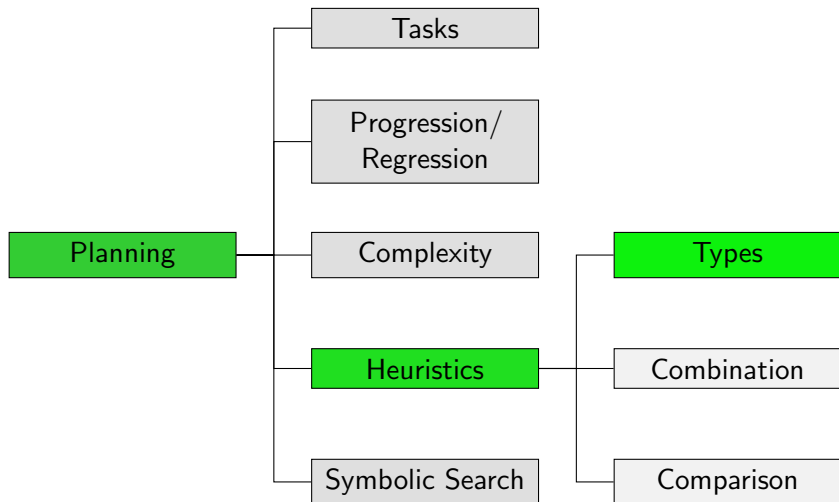
D5. Pattern Databases: Multiple Patterns

Malte Helmert and Gabriele Röger

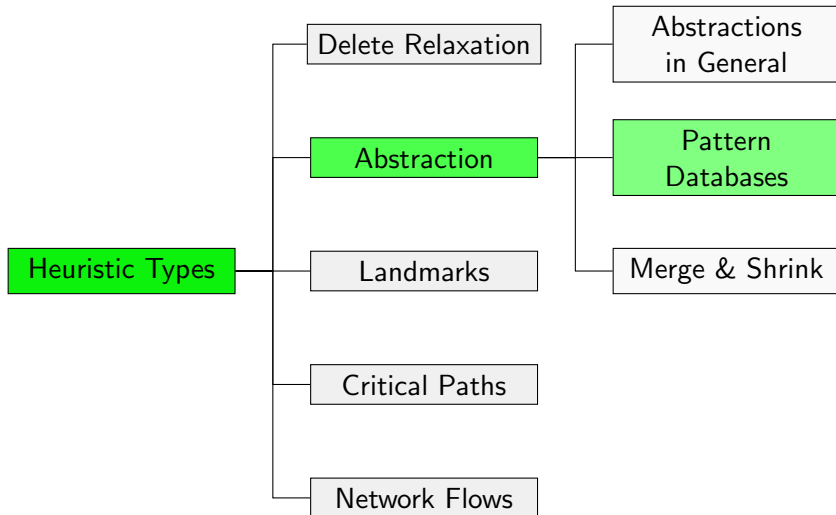
Universität Basel

November 8, 2017

Content of this Course



Content of this Course: Heuristic Types



Additivity & the Canonical Heuristic

Pattern Collections

- The space requirements for a pattern database grow **exponentially** with the **number of state variables** in the pattern.
- This places severe limits on the usefulness of single PDB heuristics h^P for larger planning task.
- To overcome this limitation, planners using pattern databases work with **collections of multiple patterns**.
- When using two patterns P_1 and P_2 , it is always possible to use the **maximum** of h^{P_1} and h^{P_2} as an admissible and consistent heuristic estimate.
- However, when possible, it is much preferable to use the **sum** of h^{P_1} and h^{P_2} as a heuristic estimate, since $h^{P_1} + h^{P_2} \geq \max\{h^{P_1}, h^{P_2}\}$.

Criterion for Additive Patterns

Theorem (Additive Pattern Sets)

Let P_1, \dots, P_k be patterns for an FDR planning task Π .

If there exists no operator that has an effect on a variable $v_i \in P_i$ and on a variable $v_j \in P_j$ for some $i \neq j$, then $\sum_{i=1}^k h^{P_i}$ is an admissible and consistent heuristic for Π .

Criterion for Additive Patterns

Theorem (Additive Pattern Sets)

Let P_1, \dots, P_k be patterns for an FDR planning task Π .

If there exists no operator that has an effect on a variable $v_i \in P_i$ and on a variable $v_j \in P_j$ for some $i \neq j$, then $\sum_{i=1}^k h^{P_i}$ is an admissible and consistent heuristic for Π .

Proof.

If there exists no such operator, then no label of $\mathcal{T}(\Pi)$ affects both $\mathcal{T}(\Pi)^{\pi_{P_i}}$ and $\mathcal{T}(\Pi)^{\pi_{P_j}}$ for $i \neq j$. By the theorem on affecting transition labels, this means that any two projections π_{P_i} and π_{P_j} are orthogonal. The claim follows with the theorem on additivity for orthogonal abstractions. □

A pattern set $\{P_1, \dots, P_k\}$ which satisfies the criterion of the theorem is called an **additive pattern set** or **additive set**.

Finding Additive Pattern Sets

The theorem on additive pattern sets gives us a simple criterion to decide which pattern heuristics can be admissibly added.

Given a **pattern collection** \mathcal{C} (i.e., a set of patterns), we can use this information as follows:

- 1 Build the **compatibility graph** for \mathcal{C} .
 - Vertices correspond to patterns $P \in \mathcal{C}$.
 - There is an edge between two vertices iff no operator affects both incident patterns.
- 2 Compute **all maximal cliques** of the graph.
These correspond to maximal additive subsets of \mathcal{C} .
 - Computing large cliques is an NP-hard problem, and a graph can have exponentially many maximal cliques.
 - However, there are **output-polynomial** algorithms for finding all maximal cliques (Tomita, Tanaka & Takahashi, 2004) which have led to good results in practice.

The Canonical Heuristic Function

Definition (Canonical Heuristic Function)

Let \mathcal{C} be a pattern collection for an FDR planning task.

The **canonical heuristic** $h^{\mathcal{C}}$ for pattern collection \mathcal{C} is defined as

$$h^{\mathcal{C}}(s) = \max_{\mathcal{D} \in \text{cliques}(\mathcal{C})} \sum_{P \in \mathcal{D}} h^P(s),$$

where $\text{cliques}(\mathcal{C})$ is the set of all maximal cliques in the compatibility graph for \mathcal{C} .

For all choices of \mathcal{C} , heuristic $h^{\mathcal{C}}$ is admissible and consistent.

How Good is the Canonical Heuristic Function?

- The canonical heuristic function is the **best possible** admissible heuristic we can derive from \mathcal{C} using **our additivity criterion**.
 - Even better heuristic estimates can be obtained from projection heuristics using a **more general additivity criterion** based on an idea called **cost partitioning**.
- ~> We will return to this topic in Part F.

Canonical Heuristic Function: Example

Example

Consider a planning task with state variables $V = \{v_1, v_2, v_3\}$ and the pattern collection $\mathcal{C} = \{P_1, \dots, P_4\}$ with $P_1 = \{v_1, v_2\}$, $P_2 = \{v_1\}$, $P_3 = \{v_2\}$ and $P_4 = \{v_3\}$.

There are operators affecting each individual variable, and the only operators affecting multiple variables affect v_1 and v_3 . What are the maximal cliques in the compatibility graph for \mathcal{C} ?

Canonical Heuristic Function: Example

Example

Consider a planning task with state variables $V = \{v_1, v_2, v_3\}$ and the pattern collection $\mathcal{C} = \{P_1, \dots, P_4\}$ with $P_1 = \{v_1, v_2\}$, $P_2 = \{v_1\}$, $P_3 = \{v_2\}$ and $P_4 = \{v_3\}$.

There are operators affecting each individual variable, and the only operators affecting multiple variables affect v_1 and v_3 . What are the maximal cliques in the compatibility graph for \mathcal{C} ?

Answer: $\{P_1\}$, $\{P_2, P_3\}$, $\{P_3, P_4\}$

Canonical Heuristic Function: Example

Example

Consider a planning task with state variables $V = \{v_1, v_2, v_3\}$ and the pattern collection $\mathcal{C} = \{P_1, \dots, P_4\}$ with $P_1 = \{v_1, v_2\}$, $P_2 = \{v_1\}$, $P_3 = \{v_2\}$ and $P_4 = \{v_3\}$.

There are operators affecting each individual variable, and the only operators affecting multiple variables affect v_1 and v_3 . What are the maximal cliques in the compatibility graph for \mathcal{C} ?

Answer: $\{P_1\}$, $\{P_2, P_3\}$, $\{P_3, P_4\}$

What is the canonical heuristic function $h^{\mathcal{C}}$?

Canonical Heuristic Function: Example

Example

Consider a planning task with state variables $V = \{v_1, v_2, v_3\}$ and the pattern collection $\mathcal{C} = \{P_1, \dots, P_4\}$ with $P_1 = \{v_1, v_2\}$, $P_2 = \{v_1\}$, $P_3 = \{v_2\}$ and $P_4 = \{v_3\}$.

There are operators affecting each individual variable, and the only operators affecting multiple variables affect v_1 and v_3 . What are the maximal cliques in the compatibility graph for \mathcal{C} ?

Answer: $\{P_1\}$, $\{P_2, P_3\}$, $\{P_3, P_4\}$

What is the canonical heuristic function $h^{\mathcal{C}}$?

Answer:
$$h^{\mathcal{C}} = \max \{h^{P_1}, h^{P_2} + h^{P_3}, h^{P_3} + h^{P_4}\}$$
$$= \max \{h^{\{v_1, v_2\}}, h^{\{v_1\}} + h^{\{v_2\}}, h^{\{v_2\}} + h^{\{v_3\}}\}$$

Dominated Additive Sets

Computing h^c Efficiently: Motivation

Consider $h^c = \max \{h^{\{v_1, v_2\}}, h^{\{v_1\}} + h^{\{v_2\}}, h^{\{v_2\}} + h^{\{v_3\}}\}$.

- We need to evaluate this expression for **every search node**.
- It is thus worth to spend some effort in precomputations to make the evaluation **more efficient**.

A naive implementation requires **4 PDB lookups** (one for each pattern) and maximizes over **3 additive sets**.

Can we do better?

Dominated Sum Theorem

Theorem (Dominated Sum)

Let $\{P_1, \dots, P_k\}$ be an additive pattern set for an FDR planning task Π , and let P be a pattern with $P_i \subseteq P$ for all $i \in \{1, \dots, k\}$. Then $\sum_{i=1}^k h^{P_i} \leq h^P$.

Dominated Sum Theorem

Theorem (Dominated Sum)

Let $\{P_1, \dots, P_k\}$ be an additive pattern set for an FDR planning task Π , and let P be a pattern with $P_i \subseteq P$ for all $i \in \{1, \dots, k\}$. Then $\sum_{i=1}^k h^{P_i} \leq h^P$.

Proof.

Because $P_i \subseteq P$, all projections π_{P_i} are **coarsenings** of the projection π_P . Let $\mathcal{T}' := \mathcal{T}(\Pi)^{\pi_P}$.

We can view each h^{P_i} as an abstraction heuristic for solving \mathcal{T}' .

By the argumentation of the previous theorem, $\{P_1, \dots, P_k\}$ is an additive pattern set and hence $\sum_{i=1}^k h^{P_i}$ is an **admissible heuristic** for solving \mathcal{T}' . Hence, $\sum_{i=1}^k h^{P_i}$ is bounded by the optimal goal distances in \mathcal{T}' , which implies $\sum_{i=1}^k h^{P_i} \leq h^P$.



Dominated Sum Corollary

Corollary (Dominated Sum)

Let $\{P_1, \dots, P_n\}$ and $\{Q_1, \dots, Q_m\}$ be additive pattern sets of an FDR planning task such that each pattern P_i is a subset of some pattern Q_j (not necessarily proper).

Then $\sum_{i=1}^n h^{P_i} \leq \sum_{j=1}^m h^{Q_j}$.

Dominated Sum Corollary

Corollary (Dominated Sum)

Let $\{P_1, \dots, P_n\}$ and $\{Q_1, \dots, Q_m\}$ be additive pattern sets of an FDR planning task such that each pattern P_i is a subset of some pattern Q_j (not necessarily proper).

Then $\sum_{i=1}^n h^{P_i} \leq \sum_{j=1}^m h^{Q_j}$.

Proof.

$$\sum_{i=1}^n h^{P_i} \stackrel{(1)}{\leq} \sum_{j=1}^m \sum_{P_i \subseteq Q_j} h^{P_i} \stackrel{(2)}{\leq} \sum_{j=1}^m h^{Q_j},$$

where (1) holds because each P_i is contained in some Q_j and (2) follows from the dominated sum theorem. □

Dominance Pruning

- We can use the dominated sum corollary to simplify the representation of h^C :
sums that are dominated by other sums can be pruned.
- The dominance test can be performed in polynomial time.

Example

$$\begin{aligned} & \max \{h^{\{v_1, v_2\}}, h^{\{v_1\}} + h^{\{v_2\}}, h^{\{v_2\}} + h^{\{v_3\}}\} \\ &= \max \{h^{\{v_1, v_2\}}, h^{\{v_2\}} + h^{\{v_3\}}\} \end{aligned}$$

↪ number of PDB lookups reduced from 4 to 3;
number of additive sets reduced from 3 to 2

Redundant Patterns

Redundant Patterns

- The previous example shows that sometimes, not all patterns in a pattern collection are **useful**.
 - Pattern $\{v_1\}$ could be removed because it does not affect the heuristic value.
 - In this section, we will show that certain patterns are **never** useful and should thus **never** be considered.
 - Knowing about such **redundant** patterns is useful for algorithms that try to find good patterns automatically.
- ~> It allows us to focus on the useful ones.

Non-Goal Patterns

Theorem (Non-Goal Patterns are Trivial)

Let Π be a SAS⁺ planning task that is not trivially unsolvable, and let P be a pattern for Π such that no variable in P is mentioned in the goal formula of Π .

Then $h^P(s) = 0$ for all states s .

Proof.

All states in the abstraction are goal states. □

↗ Patterns with no goal variables are redundant.
They should not be included in a pattern collection.

Causal Graphs: Motivation

- For more interesting notions of redundancy, we need to introduce **causal graphs**.
- Causal graphs describe the **dependency structure** between the **state variables** of a planning task.
- Causal graphs are a general tool for analyzing planning tasks.
- They are used in many contexts besides abstraction heuristics.

Causal Graphs

Definition (Causal Graph)

Let $\Pi = \langle V, I, O, \gamma \rangle$ be an FDR planning task.

The **causal graph** of Π , written **CG**(Π), is the directed graph whose vertices are the state variables V and which has an arc $\langle u, v \rangle$ iff $u \neq v$ and there exists an operator $o \in O$ such that:

- u appears anywhere in o (in precondition, effect conditions or atomic effects), and
- v is modified by an effect of o .

Idea: an arc $\langle u, v \rangle$ in the causal graph indicates that variable u is in some way relevant for modifying the value of v

Causally Relevant Variables

Definition (Causally Relevant)

Let $\Pi = \langle V, I, O, \gamma \rangle$ be an FDR planning task, and let $P \subseteq V$ be a pattern for Π .

We say that $v \in P$ is **causally relevant for P** if $CG(\Pi)$, restricted to the variables of P , contains a directed path from v to a variable $v' \in P$ that is mentioned in the goal formula γ .

Note: The definition implies that variables in P mentioned in the goal are always causally relevant for P .

Causally Irrelevant Variables are Useless

Theorem (Causally Irrelevant Variables are Useless)

Let $P \subseteq V$ be a pattern for an FDR planning task Π , and let $P' \subseteq P$ consist of all variables that are causally relevant for P . Then $h^P(s) = h^{P'}(s)$ for all states s .

↪ Patterns P where not all variables are causally relevant are redundant. The smaller subpattern P' should be used instead.

Causally Irrelevant Variables are Useless: Proof

Proof Sketch.

(\geq) : holds because π_P is a refinement of $\pi_{P'}$

Causally Irrelevant Variables are Useless: Proof

Proof Sketch.

(\geq): holds because π_P is a refinement of $\pi_{P'}$

(\leq): Obvious if $h^{P'}(s) = \infty$; else, consider an optimal abstract plan $\langle o_1, \dots, o_n \rangle$ for $\pi_{P'}(s)$ in $\mathcal{T}(\Pi)^{\pi_{P'}}$.

W.l.o.g., each o_i modifies some variable in P' .

(Other o_i are redundant and can be omitted.)

Because P' includes all variables causally relevant for P , no variable in $P \setminus P'$ is mentioned in any o_i or in the goal.

Then the same abstract plan also is a solution for $\pi_P(s)$ in $\mathcal{T}(\Pi)^{\pi_P}$.

Hence, the optimal solution cost under abstraction π_P is no larger than under $\pi_{P'}$.

Causally Connected Patterns

Definition (Causally Connected)

Let $\Pi = \langle V, I, O, \gamma \rangle$ be an FDR planning task, and let $P \subseteq V$ be a pattern for Π .

We say that P is **causally connected** if the subgraph of $CG(\Pi)$ induced by P is weakly connected (i.e., contains a path from every vertex to every other vertex, ignoring arc directions).

Disconnected Patterns are Decomposable

Theorem (Causally Disconnected Patterns are Decomposable)

Let $P \subseteq V$ be a pattern for a SAS^+ planning task Π that is not causally connected, and let P_1, P_2 be a partition of P into non-empty subsets such that $CG(\Pi)$ contains no arc between the two sets.

Then $h^P(s) = h^{P_1}(s) + h^{P_2}(s)$ for all states s .

↪ Causally disconnected patterns P are redundant.

The smaller subpatterns P_1 and P_2 should be used instead.

Disconnected Patterns are Decomposable: Proof

Proof Sketch.

(\geq): There is no arc between P_1 and P_2 in the causal graph, and thus there is no operator that affects both patterns.

Therefore, they are additive, and $h^P \geq h^{P_1} + h^{P_2}$ follows from the dominated sum theorem.

(\leq): Obvious if $h^{P_1}(s) = \infty$ or $h^{P_2}(s) = \infty$. Else, consider optimal abstract plans ρ_1 for $\mathcal{T}(\Pi)^{\pi_{P_1}}$ and ρ_2 for $\mathcal{T}(\Pi)^{\pi_{P_2}}$.

Because the variables of the two projections do not interact, concatenating the two plans yields an abstract plan for $\mathcal{T}(\Pi)^{\pi_P}$.

Hence, the optimal solution cost under abstraction π_P is at most the sum of costs of ρ_1 and ρ_2 , and thus $h^P \leq h^{P_1} + h^{P_2}$.

Summary

Summary (1)

- When faced with multiple PDB heuristics (a **pattern collection**), we want to **admissibly add** their values where possible, and **maximize** where addition is inadmissible.
- A set of patterns is **additive** if each operator affects (i.e., assigns to a variable from) at most one pattern in the set.
- The **canonical heuristic function** is the **best possible** additive/maximizing combination for a given pattern collection given this additivity criterion.

Summary (2)

Not all patterns need to be considered, as some are **redundant**:

- Patterns should include a **goal variable** (else $h^P = 0$).
- Patterns should only include **causally relevant** variables (others can be dropped without affecting the heuristic value).
- Patterns should be **causally connected** (disconnected patterns can be split into smaller subpatterns at no loss).