Additive Pattern Databases for Decoupled Search

Silvan Sievers\textsuperscript{1}  Daniel Gnad\textsuperscript{2}  Álvaro Torralba\textsuperscript{3}

\textsuperscript{1}University of Basel, Switzerland
\textsuperscript{2}Linköping University, Sweden
\textsuperscript{3}Aalborg University, Denmark

HSDIP, June 2022
Motivation

- state of the art in optimal classical planning: explicit heuristic search with abstractions
- goal: use abstraction heuristics also with decoupled search
- contribution: pattern database (PDB) and their additive combination for decoupled search
Background

- planning tasks: finite-domain **state variables** for representing states
Background

- planning tasks: finite-domain state variables for representing states
- pattern database (PDB) heuristics:
  - project variables to a subset
  - store perfect heuristic values of abstraction
partition state variables to decompose the task: center factor and leaf factors
branch over center states and actions, handle leaves separately
Decoupled Search in a Nutshell

- partition state variables to decompose the task: center factor and leaf factors
- branch over center states and actions, handle leaves separately
- decoupled state $s^F$:
  - center state
  - pricing function: cost of reachable leaf states
partition state variables to decompose the task: center factor and leaf factors
branch over center states and actions, handle leaves separately
decoupled state $s^F$:
- center state
- pricing function: cost of reachable leaf states
- → represents exponentially many (explicit) member states
previous approaches based on **buy-leaves compilation:**
- compile prices of current state into task
- compute arbitrary explicit heuristic on transformed task
Heuristics for Decoupled Search So Far

- previous approaches based on buy-leaves compilation:
  - compile prices of current state into task
  - compute arbitrary explicit heuristic on transformed task
- problems:
  - can be information-lossy (e.g., delete relaxation)
  - impractical for abstraction-based heuristics
explicit decoupled heuristic

\[ h_{\mathcal{F}, \text{ex}}(s^\mathcal{F}) = \min_{s \in [s^\mathcal{F}]} \text{price}(s^\mathcal{F}, s) + h(s) \]
Alternative Definition for Computing Decoupled Heuristics

explicit decoupled heuristic

$$h_{F,ex}(s^F) = \min_{s \in \mathcal{S}^F} \text{price}(s^F, s) + h(s)$$

- problem: exponentially many member states
Single PDBs for Decoupled Search


reminder: explicit decoupled heuristic

\[
 h_{\mathcal{F},\text{ex}}(s^\mathcal{F}) = \min_{s \in [s^\mathcal{F}]} \text{price}(s^\mathcal{F}, s) + h(s)
\]
Single PDBs for Decoupled Search

Remainder: explicit decoupled heuristic

\[ h_{\mathcal{F}, \text{ex}}(s^\mathcal{F}) = \min_{s \in [s^\mathcal{F}]} \text{price}(s^\mathcal{F}, s) + h(s) \]

Decoupled PDB

\[ d\text{PDB}(h^P, s^\mathcal{F}) = \min_{s^P \in S^P} \text{price}(s^\mathcal{F}, s^P) + h^P(s^P) \]
Combining Multiple PDBs

**explicit search**

- given: $\mathcal{H} = \{H_1, \ldots, H_n\}$ with $H_i$ additive set of (PDB) heuristics
- canonical combination:

$$h^\mathcal{H}(s) = \max_{H \in \mathcal{H}} \sum_{h \in H} h(s)$$
Combining Multiple PDBs

**explicit search**

- given: $\mathcal{H} = \{H_1, \ldots, H_n\}$ with $H_i$ additive set of (PDB) heuristics
- canonical combination:

  $$h^\mathcal{H}(s) = \max_{H \in \mathcal{H}} \sum_{h \in H} h(s)$$

- how to transfer to decoupled search?
Naïve Combination

reminder: canonical heuristic

$$h^H(s) = \max_{H \in H} \sum_{h \in H} h(s)$$

compute decoupled PDBs individually:

- $$h^H_F(s) = \max_{H \in H} \sum_{h \in H} h(s),$$
- naïve

properties

- information-lossy: use different minimizing member state in each PDB
- inadmissible: may count prices of leaves multiple times in different heuristics
Naïve Combination

reminder: canonical heuristic

\[ h^\mathcal{H}(s) = \max_{H \in \mathcal{H}} \sum_{h \in H} h(s) \]

▶ compute decoupled PDBs individually:

\[ h_\mathcal{F}^{\mathcal{H},\text{naïve}}(s^{\mathcal{F}}) = \max_{H \in \mathcal{H}} \sum_{h \in H} h_\mathcal{F}^{\text{ex}}(s^{\mathcal{F}}) \]
Naïve Combination

reminder: canonical heuristic

\[ h^H(s) = \max_{H \in \mathcal{H}} \sum_{h \in H} h(s) \]

- compute decoupled PDBs **individually**:

\[ h^H_{\mathcal{F}, \text{naïve}}(s^F) = \max_{H \in \mathcal{H}} \sum_{h \in H} h_{\mathcal{F}, \text{ex}}(s^F) \]

**properties**

- **information-lossy**: use different minimizing member state in each PDB
- **inadmissible**: may count prices of leaves multiple times in different heuristics
Explicit Decoupled Canonical Heuristic (1)

reminder: explicit decoupled heuristic

\[ h_{\mathcal{F}, \text{ex}}(s^\mathcal{F}) = \min_{s \in [s^\mathcal{F}]} \text{price}(s^\mathcal{F}, s) + h(s) \]
Explicit Decoupled Canonical Heuristic (1)

Reminder: explicit decoupled heuristic

\[ h_{\mathcal{F},\text{ex}}(s^\mathcal{F}) = \min_{s \in [s^\mathcal{F}]} \text{price}(s^\mathcal{F}, s) + h(s) \]

Now: \( h(s) = h^\mathcal{H}(s) = \max_{\mathcal{H} \in \mathcal{H}} \sum_{h \in \mathcal{H}} h(s) \)
Explicit Decoupled Canonical Heuristic (1)

reminder: explicit decoupled heuristic

\[ h_{\mathcal{F}, \text{ex}}(s^{\mathcal{F}}) = \min_{s \in [s^{\mathcal{F}}]} \text{price}(s^{\mathcal{F}}, s) + h(s) \]

▶ now: \( h(s) = h^\mathcal{H}(s) = \max_{H \in \mathcal{H}} \sum_{h \in H} h(s) \)

explicit decoupled canonical heuristic

\[ h^\mathcal{H}_{\mathcal{F}, \text{ex}}(s^{\mathcal{F}}) = \min_{s \in [s^{\mathcal{F}}]} \text{price}(s^{\mathcal{F}}, s) + \max_{H \in \mathcal{H}} \sum_{h \in H} h(s) \]
Explicit Decoupled Canonical Heuristic (2)

complexity

computing $h^H_{x,ex}$ is NP-complete
Explicit Decoupled Canonical Heuristic (2)

**complexity**

computing $h^H_{F, \text{ex}}$ is **NP-complete**

- practical implementation via branch-and-bound
- incremental computation of member states allows pruning
- worst case: enumeration of all exponentially many member states
Polynomial-time Approximations (1)

reminder: explicit decoupled canonical heuristic

\[ h^H_{\mathcal{F}, \text{ex}}(s^\mathcal{F}) = \min_{s \in [s^\mathcal{F}]} \text{price}(s^\mathcal{F}, s) + \max_{H \in \mathcal{H}} \sum_{h \in H} h(s) \]
reminder: explicit decoupled canonical heuristic

\[ h_{\mathcal{F}, \text{ex}}^{\mathcal{H}}(s^{\mathcal{F}}) = \min_{s \in [s^{\mathcal{F}}]} \text{price}(s^{\mathcal{F}}, s) + \max_{H \in \mathcal{H}} \sum_{h \in H} h(s) \]

- expensive part: finding minimizing member state \( s \)
remind: explicit decoupled canonical heuristic

\[ h_{\mathcal{F}, \text{ex}}(s^\mathcal{F}) = \min_{s \in [s^\mathcal{F}]} \text{price}(s^\mathcal{F}, s) + \max_{H \in \mathcal{H}} \sum_{h \in H} h(s) \]

- expensive part: finding minimizing member state \( s \)
- alternative: consider each \( H \in \mathcal{H} \) independently, i.e., swap min and max:

\[ \max_{H \in \mathcal{H}} \min_{s \in [s^\mathcal{F}]} \text{price}(s^\mathcal{F}, s) + \sum_{h \in H} h(s) \]
Polynomial-time Approximations (1)

reminder: explicit decoupled canonical heuristic

\[ h_{\mathcal{F}, \text{ex}}^H(s^F) = \min_{s \in [s^F]} \text{price}(s^F, s) + \max_{H \in \mathcal{H}} \sum_{h \in H} h(s) \]

▶ expensive part: finding minimizing member state \( s \)
▶ alternative: consider each \( H \in \mathcal{H} \) independently, i.e., swap min and max:

\[ \max_{H \in \mathcal{H}} \min_{s \in [s^F]} \text{price}(s^F, s) + \sum_{h \in H} h(s) \]

▶ admissible, but lossy approximation
Leaf-Disjoint PDBs

additive sets: pairwise leaf-disjoint PDBs
Leaf-Disjoint PDBs
additive sets: pairwise leaf-disjoint PDBs

Single-Leaf PDBs
each PDB affects at most one leaf
Leaf-Disjoint PDBs
additive sets: pairwise leaf-disjoint PDBs

Single-Leaf PDBs
each PDB affects at most one leaf

▶ minimize sum of prices and heuristic separately for each set of affected leaves
▶ heuristic value equals $h^H_{\mathcal{F},\text{ex}}$
## Experiments

<table>
<thead>
<tr>
<th></th>
<th>F</th>
<th>MM</th>
</tr>
</thead>
<tbody>
<tr>
<td>decoupled</td>
<td>284</td>
<td>749</td>
</tr>
<tr>
<td>explicit</td>
<td>206</td>
<td>662</td>
</tr>
<tr>
<td>expl. dec. heur.</td>
<td>293</td>
<td>743</td>
</tr>
<tr>
<td>poly. approx.</td>
<td>206</td>
<td>596</td>
</tr>
<tr>
<td>LD</td>
<td>212</td>
<td>628</td>
</tr>
<tr>
<td>SL</td>
<td>210</td>
<td>607</td>
</tr>
<tr>
<td>no pruning</td>
<td>304</td>
<td>707</td>
</tr>
<tr>
<td>pruning</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Experiments

<table>
<thead>
<tr>
<th>Method</th>
<th>F</th>
<th>MM</th>
<th>LD</th>
<th>SL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Explicit search</td>
<td>284</td>
<td>662</td>
<td>206</td>
<td>749</td>
</tr>
</tbody>
</table>

The table above compares the performance of explicit search under different conditions. The values represent some metric or measure, with F and MM indicating different scenarios or methods, and LD and SL representing different configurations or parameters.
Experiments

<table>
<thead>
<tr>
<th></th>
<th>explicit search</th>
<th>decoupled search</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>LD</td>
<td>SL</td>
</tr>
<tr>
<td>F</td>
<td>284</td>
<td>206</td>
</tr>
<tr>
<td>MM</td>
<td>749</td>
<td>662</td>
</tr>
</tbody>
</table>
## Experiments

<table>
<thead>
<tr>
<th></th>
<th>explicit search</th>
<th>decoupled search</th>
<th>expl. dec. heur.</th>
<th>no pruning</th>
<th>pruning</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>LD</td>
<td>SL</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>284</td>
<td>206</td>
<td>293</td>
<td>206</td>
<td>212</td>
</tr>
<tr>
<td>MM</td>
<td>749</td>
<td>662</td>
<td>743</td>
<td>596</td>
<td>628</td>
</tr>
</tbody>
</table>
## Experiments

<table>
<thead>
<tr>
<th></th>
<th>explicit search</th>
<th></th>
<th>expl. dec. heur.</th>
<th>poly. approx.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>LD</td>
<td>SL</td>
<td>no pruning</td>
<td>pruning</td>
</tr>
<tr>
<td>F</td>
<td>284</td>
<td>206</td>
<td>293</td>
<td>206</td>
</tr>
<tr>
<td>MM</td>
<td><strong>749</strong></td>
<td>662</td>
<td>743</td>
<td>596</td>
</tr>
</tbody>
</table>
### Experiments

<table>
<thead>
<tr>
<th></th>
<th>explicit search</th>
<th>decoupled search</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>LD</td>
<td>SL</td>
<td>expl. dec. heur. no pruning</td>
</tr>
<tr>
<td>F</td>
<td>284</td>
<td>206</td>
<td>206</td>
</tr>
<tr>
<td>MM</td>
<td><strong>749</strong></td>
<td>662</td>
<td>596</td>
</tr>
</tbody>
</table>
Conclusions

- alternative way of computing explicit heuristics for decoupled search
- efficient computation of PDBs for decoupled search
- admissible combination of sets of additive PDBs NP-complete
- practical implementation and polynomial-time approximations