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

SoCS, 22nd July 2022
Setting & Motivation

- optimal classical planning as heuristic search
- state of the art: abstraction heuristics
- successful alternative to explicit search: decoupled search
Setting & Motivation

- optimal classical planning as heuristic search
- state of the art: abstraction heuristics
- successful alternative to explicit search: decoupled search
- goal: abstraction heuristics for decoupled search
Background

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

- planning tasks: finite-domain \textit{state variables} for representing states
- pattern database (PDB) heuristics:
  - project variables to a \textit{subset}
  - store perfect heuristic values of abstraction
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 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
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
  - represents exponentially many (explicit) member states
Heuristics for Decoupled Search So Far

- given: explicit heuristic $h$
- given: decoupled state $s^F$
- question: how to use $h$?
Heuristics for Decoupled Search So Far

- given: explicit heuristic $h$
- given: decoupled state $s^F$
- question: how to use $h$?

**buy-leaves compilation**

- compile prices of $s^F$ into new task
- evaluate $h$ on compiled task
Heuristics for Decoupled Search So Far

- given: explicit heuristic \( h \)
- given: decoupled state \( s^F \)
- question: how to use \( h \)?

**buy-leaves compilation**

- compile prices of \( s^F \) into new task
- evaluate \( h \) on compiled task

**problems:**
- impractical for abstraction-based heuristics
- pattern selection based on original task
contribution: explicit decoupled heuristic

\[ h_{F,\text{ex}}(s^F) = \min_{s \in [s^F]} \text{price}(s^F, s) + h(s) \]
contribution: explicit decoupled heuristic

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

- “best” use of given explicit heuristic
- 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

Reminder: explicit decoupled heuristic

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

Contribution: 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 (e.g., disjoint PDBs, cost-partitioned PDBs, etc.)

- **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 (e.g., disjoint PDBs, cost-partitioned PDBs, etc.)
- 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

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

**reminder: canonical heuristic**

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

▶ evaluate PDBs **individually** (use dPDB to compute \(h_{\mathcal{F},\text{ex}}(s^\mathcal{F})\)):

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

reminder: canonical heuristic

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

- evaluate PDBs individually (use dPDB to compute \( h_{\mathcal{F},\text{ex}}(s^\mathcal{F}) \)):

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

properties

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

reminder: explicit decoupled heuristic

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

Contribution: 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 $h^H_{\mathcal{F}, \text{ex}}$ is NP-complete
Explicit Decoupled Canonical Heuristic (2)

Computing $h^H_{F,ex}$ 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
remind me: explicit decoupled canonical heuristic

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

reminder: explicit decoupled canonical heuristic

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

▶ alternative: consider each \( H \in \mathcal{H} \) independently, i.e., move max outward:

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

reminder: explicit decoupled canonical heuristic

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

▶ alternative: consider each \( H \in \mathcal{H} \) independently, i.e., move max outward:

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

▶ admissible, but lossy approximation
<table>
<thead>
<tr>
<th>leaf-disjoint (LD) PDBs</th>
</tr>
</thead>
<tbody>
<tr>
<td>additive sets: pairwise leaf-disjoint PDBs</td>
</tr>
</tbody>
</table>
leaf-disjoint (LD) PDBs
additive sets: pairwise leaf-disjoint PDBs

single-leaf (SL) PDBs
each PDB affects at most one leaf
Polynomial-time Approximations (2)

leaf-disjoint (LD) PDBs
additive sets: pairwise leaf-disjoint PDBs

single-leaf (SL) 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

- PDBs computed with hill climbing and CEGAR
- additivity obtained through saturated cost partitioning
Experiments

- PDBs computed with hill climbing and CEGAR
- Additivity obtained through saturated cost partitioning
- Coverage:

\[
\begin{array}{cccc}
F & 284 & 206 & 293 \\
MM & 749 & 662 & 743
\end{array}
\]

\[
\begin{array}{cccc}
LD & 212 & 210 & 304 \\
SL & 200 & 200 & 200
\end{array}
\]
Experiments

- PDBs computed with hill climbing and CEGAR
- Additivity obtained through saturated cost partitioning
- Coverage:

<table>
<thead>
<tr>
<th></th>
<th>explicit search</th>
<th></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

- PDBs computed with hill climbing and CEGAR
- Additivity obtained through saturated cost partitioning
- Coverage:

<table>
<thead>
<tr>
<th></th>
<th>Explicit Search</th>
<th>Decoupled Search</th>
</tr>
</thead>
<tbody>
<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

- PDBs computed with hill climbing and CEGAR
- Additivity obtained through saturated cost partitioning
- Coverage:

<table>
<thead>
<tr>
<th></th>
<th>Explicit Search</th>
<th>Decoupled Search</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>284</td>
<td>206</td>
</tr>
<tr>
<td>MM</td>
<td><strong>749</strong></td>
<td>662</td>
</tr>
</tbody>
</table>
Experiments

- PDBs computed with hill climbing and CEGAR
- Additivity obtained through saturated cost partitioning
- Coverage:

<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>
Conclusions

- **summary:**
  - alternative way of computing explicit heuristics for decoupled search
  - efficient computation of PDBs for decoupled search
  - admissible combination of sets of additive PDBs \textit{NP-complete}
  - practical implementation and \textit{polynomial-time approximations}

- future work:
  - many results independent of type of heuristic: use different abstractions
  - integrate cost partitioning into decoupled search: leaf price partitioning
Conclusions

summary:
- 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

future work:
- many results independent of type of heuristic: use different abstractions
- integrate cost partitioning into decoupled search: leaf price partitioning