Additive Pattern Databases for Decoupled Search: Additional Material

Silvan Sievers,1 Daniel Gnad,2 Álvaro Torralba3

1 University of Basel, Switzerland
2 Linköping University, Sweden
3 Aalborg University, Denmark

silvan.sievers@unibas.ch, daniel.gnad@liu.se, alto@cs.aau.dk

This additional material contains full proofs for all claims of the paper (Sievers, Gnad, and Torralba 2022) with a proof sketch or without proof. It further contains full per-domain coverage results for all reported experiments. Finally, it also reports all experimental results of the main paper on the Autoscale 21.11 benchmarks (Torralba, Seipp, and Sievers 2021).

Proposition 2. Let \( h, h^{LS} \) be two consistent and buy-leaves agnostic heuristics. Then, for every decoupled state \( s^F \) it holds that \( h_{\mathcal{F},\text{ex}}(s^F) \geq h_{\mathcal{F},\text{comp}}(s^F) \).

Proof. Let \( s \in [s^F] \) be the state that minimizes \( h_{\mathcal{F},\text{ex}}(s^F) = \min_{\pi \in [s^F]} \text{price}(s^F, s) + h(s) \). Then the sequence of corresponding buy actions \( a[s[L]] \) is a path in \( \Pi_{LS} \) that is applicable in \( s^F_{LS} \) and ends in \( s^{LS} = s \cup \{ \text{bought}[L] = \top \mid L \in \mathcal{L} \} \).

First, as argued in the proof of Proposition 1 of the main paper, every plan for \( \Pi_{LS} \) can be reordered such that all \( |\mathcal{L}| \) buy actions \( a[s[L]] \) are moved to the front. Therefore, it suffices to consider only those plans for the compilation that start with any combination of such actions, which we do in the following.

Second, we show that \( h^{LS}(s^F_{LS}) \leq \text{price}(s^F, s) + h^{LS}(s^{LS}) \). This holds because the cost of the sequence of actions \( a[s[L]] \) that leads from \( s^F_{LS} \) to \( s^{LS} \) is exactly \( \text{price}(s^F, s) \) and, because \( h^{LS} \) is consistent, the heuristic cannot decrease by more than the cost of this path.

Putting the pieces together, we get the claim as follows:

\[
  h_{\mathcal{F},\text{ex}}(s^F) = \text{price}(s^F, s) + h(s)
  = \text{price}(s^F, s) + h^{LS}(s^{LS})
  \geq h^{LS}(s^{LS}) = h_{\mathcal{F},\text{comp}}(s^F)
\]

The first and last equalities hold by the definitions of \( h_{\mathcal{F},\text{ex}} \), respectively \( h_{\mathcal{F},\text{comp}} \), the second one because \( (h, h^{LS}) \) is buy-leaves agnostic, and the inequality because \( h^{LS} \) is consistent.

Proposition 4. Let \( \Pi \) be a planning task with factoring \( \mathcal{F} \), and \( h^{\mathcal{F}} \) a PDB heuristic for \( \Pi \). There exists a heuristic \( h^{\mathcal{F}}_{\text{comp}} \) for the buy-leaves compilation \( \Pi_{LS} \) such that for all decoupled states \( s^F \): \( h^{\mathcal{F}}_{\text{comp}}(s^F) = h_{\mathcal{F},\text{ex}}(s^F) \).

Proof. We set \( P' = P \cup \{ \text{bought}[L] \mid L \in \mathcal{L} \} \) and show that abstract solutions \( \pi^{\mathcal{F}} \) for \( \Pi_{LS} \) (which minimize \( h_{\mathcal{F},\text{ex}}(s^F) = \min_{\pi \in [s^F]} \text{price}(s^F, s) + h^{\mathcal{F}}(s) \)) are in one-to-one correspondence with abstract solutions \( \pi^{P'} \) for \( h^{P'} \) in \( \Pi_{LS} \), which shows the claim. The selection of leaf states in \( \text{price}(s^F, s) \) directly corresponds to the subsequence of actions \( a[s[L]] \) of a plan \( \pi^{P'} \) because incorporating \( \text{bought}[L] \) in \( P' \) forces the PDB to include exactly one action \( a[s[L]] \) for each \( L \). The remaining actions in \( \pi^{P'} \) are part of \( \Pi \) possibly with a precondition \( \text{bought}[L] = \top \). Rearranging \( \pi^{P'} \), as in the proof of Proposition 2, so that all \( a[s[L]] \) actions are in front, it remains to show that \( h^{P'}(s) = h^{P'}(s \cup \{ \text{bought}[L] = \top \mid L \in \mathcal{L} \}) \). This is true because the abstract state spaces below these two states are isomorphic, as \( P' \) includes the \( \text{bought}[L] \) variables, so exactly the same actions are applicable and the descendant states are identical except for the no-longer relevant \( \text{bought}[L] = \top \) facts.

Proposition 5. \( h^{P}_{\mathcal{F},\text{comp}}(s^F) \leq h^{P}_{\mathcal{F},\text{ex}}(s^F) \).

Proof. Let \( \Pi \) be a planning task, \( \mathcal{F} \) a factoring, and \( h^{P} \) a PDB heuristic for \( \Pi \). Let further \( s \) be the member state that minimizes \( h_{\mathcal{F},\text{ex}}(s^F) = \min_{\pi \in [s^F]} \text{price}(s^F, s) + h^{\mathcal{F}}(s) \). This corresponds to an abstract solution \( \pi^{P} \) for \( h^{P} \) in \( \Pi_{LS} \) that takes the buy actions \( a[s[L]] \) for each \( L \) followed by the abstract path corresponding to \( h^{P}(s) \). The abstract state spaces of \( h^{P} \) on \( \Pi_{LS} \) differ only in the additional actions \( a[s[L]] \) of \( \Pi_{LS} \). Thus, every action sequence \( \pi^{P} \) is also an abstract solution for \( h^{P} \) in \( \Pi_{LS} \). The other way around, since an abstract solution for \( \Pi_{LS} \) can include an arbitrary number of actions \( a[s[L]] \) for each \( L \), not every abstract solution for \( \Pi_{LS} \) corresponds to a solution for \( h_{\mathcal{F},\text{ex}}(s^F) \), proving the claim.

Proposition 6. \( h^{H}_{\mathcal{F},\text{max}} \leq h^{H}_{\mathcal{F},\text{ex}} \) and there are cases where the inequality is strict.

Proof. We show that \( h^{H}_{\mathcal{F},\text{max}} \) and \( h^{H}_{\mathcal{F},\text{max}} \) differ only on the order of the min and max operations, and the claim then follows with the max-min inequality. As all elements of \( H \) are singletons, we can drop the sum \( \sum_{h \in H} \) Then \( h^{H}_{\mathcal{F},\text{ex}}(s^F) = \min_{\pi \in [s^F]} \text{price}(s^F, s) + \max_{h \in H} h(s) \) and since the price of \( s \) is independent of \( H \) we can move
it into the max. For $h^*$ we have: $h^*$ we have: $h^*$

To see that the inequality is strict, consider a case where we take the maximum of two PDB heuristics $h_1$ and $h_2$, on a decoupled state $s^F$ with two member states $[s^F] = \{s_1, s_2\}$ with price 0. Assume $h_1(s_1) = 0$, $h_1(s_2) = \infty$, and $h_2(s_1) = \infty$, $h_2(s_2) = 0$. Then, $h^*_{\text{ex}}(s^F) = 0$, as the minimum of each heuristic is 0. However, both member states are detected as dead-ends by one heuristic so $h^*_{\text{ex}}(s^F) = \infty$.

**Theorem 1.** The Decoupled Additive PDBs problem is NP-complete.

**Proof.** Membership: Guessing a member state $s \in [s^F]$, test whether $\text{price}(s^F, s) + h^*_{\text{ex}}(s) \leq B$ in polynomial time.

**Hardness:** Reduction from 3-SAT. Given any 3-CNF formula $\phi$ with Boolean propositions $X$ and clauses $\{C_1, \ldots, C_m\}$, we construct a planning task, and decoupled state $s^F$ such that $h^*_{\text{ex}}(s^F) = 0$ if $\phi$ is satisfiable, and $h^*_{\text{ex}}(s^F) = \infty$ otherwise.

The (simplified) $H$ has the following variables:

- For each clause $C_i$ over propositions $x, y, z \in X$, a leaf $L_i$ with 4 variables $L_i = \{v_i^z, v_i^y, v_i^y, v_i^v\}$, $v_i^z$ has domain $\{a, b\}$, with value $a$ in the initial state. The rest have domain $\{0, 1\}$, and have value $u$ in the initial state.
- A center variable $v_x$ for each proposition $x \in X$, plus an extra variable $v_{v_{\text{center}}}$, all with domain $\{i, g\}$, value $i$ in the initial state and value $g$ in the goal.

The set of actions in $H$, and the pattern collection ensure that:

1. In the initial decoupled state, all leaf states in $L_i$ of the form $b_{x,y}a_{x,y}a_{z}$ are reachable with price 0 where $a_{x,y}a_{z}$ correspond to an assignment over propositions $x, y$, and $z$ that satisfy $C_i$. Therefore prices $(F^L_j)[L_i] = 0$ iff $L_i$ satisfies the clause $C_i$, else prices $(F^L_j)[L_i]$.  

2. Some state $s^F = F^L_{a_{\text{center}}}$ is reachable where $v_i = b$ for all $i \in [m]$, and prices $(s^F)[s^L_i] = \infty$ if $s^L_i[v_\text{center}] = u$. Therefore, no variable is allowed to have undefined value, and actions changing the assignment are no longer applicable. Member states in $s^F$ correspond exactly to assignments that satisfy all clauses, though they may assign different values to propositions in different leaves.

3. For any $x \in X$ appearing in clauses $C_i$ and $C_j$, the collection has a PDB $P_{x,i,j}$ such that $h^P_{x,i,j}(s) = \infty$ for any state $s$ in which $v_{i,x} \neq v_{i,y}$. Therefore, all assignments that are inconsistent are detected as dead-end by at least one PDB.

Note that (1-3) together imply that a member state $s \in [s^F]$ corresponds to a satisfying assignment if and only if $\text{price}(s^F, s) + \sum_{x,i,j} h^P_{x,i,j}(s) = 0$ and otherwise the value is $\infty$. (1) and (2) make sure that $\text{price}(s^F)[s^L_i] = 0$ iff variables $v_i^z$ correspond to a satisfying assignment (i.e., they are compatible with $C_i$ and they do not have value $u$). However, there are separate variables for each occurrence of the proposition in each clause. Then, by (3), we construct a collection of PDB heuristics such that, all states having inconsistent assignments (assigning true and false to the same proposition) have a heuristic value of $\infty$. Therefore, $h^*_{\text{ex}}(s^F) < B$ if and only if there exists a satisfying assignment to $\phi$.

For (1), we introduce 0-cost leaf actions with precondition $v_i^z = v_i^y = v_i^y = u$, $v_i^v = a$, and effect $v_i^z = a_x$, $v_i^y = a_y$, $v_i^v = a_z$, $v_i^v = b$, where $a_x, a_y, a_z$ corresponds to any assignment of $x, y, z$ that satisfies $C_i$ (7 actions in total, one per assignment). Figure 1 shows the reachable leaf states induced by these actions. Note that all leaf states have price of 0 in the initial state.

For (2), we introduce a center action $a_{\text{center}}$ with precondition $v_{\text{center}} = i \cup \{v_i^z = b \mid i \in [m]\}$ and effect $v_{\text{center}} = g$. $s^F$ is the state reached from $F^L_{a_{\text{center}}}$ by applying $a_{\text{center}}$. Note that the only change to the pricing function is that the leaf state auuu has a price of $\infty$ in $s^F$ due to not fulfilling the precondition $v_i^z = b$. All other leaf states still have a price of 0.

For (3), we introduce center actions set $(x, a_x)$ for $x \in X$ and $a_x \in \{0, 1\}$ with precondition $v_x = i \cup \{v_{i,x}^v = a_x \mid i \in C_i\}$, and effect $v_x = g$. For each $x$, one of these two actions must be applied, and this is only possible for states where all $v_{i,x}$ agree on the value of $x$. Therefore, any state in which $v_{i,x}^v \neq v_{i,x}^v$ is a dead-end state.

Finally, our PDB collection has a PDB for each center variable and each pair of clauses $i, j$, $P = \{v_i^z, v_i^v, v_x\}$. Figure 2 shows the reachable abstract state space of one PDB. There are exactly two abstract states detected as dead-end: where $u^1 = 0$ and $u^2 = 1$ and vice versa. Note that, together, these PDBs detect all dead-end states.

It is worth noting that, even though our PDBs are not orthogonal they can still be additive through cost-partitioning.

![Figure 1: Example of the leaf state space for leaf $L_1 = \{v_1^z, v_1^v, v_1^v, v_1^v\}$ corresponding to clause $x \lor y \lor \neg z$.](image1.png)

![Figure 2: Reachable abstract state space of PDB with $\{v_1^z, v_2^z, v_x\}$](image2.png)
(a) Search time of decoupled search with a single PDB heuristic, using explicit decoupled heuristic (EXP) vs. decoupled PDB (dPDB) MM, comparing using unrestricted PDBs (fE) vs. PDBs restricted to be leaf-disjoint (LD; left) or to affect a single leaf (SL; right).

(b) Expansions of explicit search with the SCP heuristic and factoring explicit decoupled heuristic (EXP) vs. decoupled PDB (dPDB) MM, comparing using unrestricted PDBs (fE) vs. PDBs restricted to be leaf-disjoint (LD; left) or to affect a single leaf (SL; right).

(c) Expansions (2 plots on the left) and search time (2 plots on the right) of decoupled search with the SCP heuristic and factoring MM, comparing explicit decoupled heuristic computation (EXP) vs. leaf-disjoint (LD; left) and single-leaf (SL; right) approximations.

Figure 3: Like Figure 2 in the main paper but on Autoscale 21.11 benchmarks.

Table 1: Like Table 1 in the main paper but on Autoscale 21.11 benchmarks.

<table>
<thead>
<tr>
<th></th>
<th>explicit search (E)</th>
<th>EXP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>p</td>
<td>f</td>
</tr>
<tr>
<td>p</td>
<td>F 86 86 86 71</td>
<td>65</td>
</tr>
<tr>
<td></td>
<td>IF 92 91 91 80</td>
<td>82</td>
</tr>
<tr>
<td></td>
<td>MM 345 327 327 304</td>
<td>269</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SCP</td>
<td>p</td>
</tr>
<tr>
<td></td>
<td>F 127 124 90 123</td>
<td>71</td>
</tr>
<tr>
<td></td>
<td>IF 108 108 102 105</td>
<td>93</td>
</tr>
<tr>
<td></td>
<td>MM 399 378 337 349</td>
<td>288</td>
</tr>
</tbody>
</table>

Table 1 and Figure 3 show the same results as Table 1 and Figure 2 in the main paper, respectively, but on the Autoscale 21.11 benchmarks (Torralba, Seipp, and Sievers 2021). Tables 2 and 3 show the full per-domain coverage results on the IPC and Autoscale benchmarks.

References
Table 2: Full per-domain coverage on the IPC benchmarks (Table 1 in the main paper).
Table 3: Full per-domain coverage on the Autoscale 21.11 benchmarks (Table 1 in this document).