In a Nutshell

- Optimal classical planning
- A* search with admissible heuristic
- Multiple heuristics capture different aspects of task
- Beneficial to combine information of these heuristics
- Cost partitioning allows admissible combination
- Greedy method: saturated cost partitioning
- Contribution: combine two orthogonal generalizations

Saturated Cost Partitioning (SCP)

Saturated cost partitioning algorithm

for heuristic \( h \) in sequence \( h_1, \ldots, h_n \) do
  \( \text{oef}_i \leftarrow \text{saturate}(h, \text{oef}) \)
  \( \text{oef} \leftarrow \text{oef} - \text{oef}_i \)
end for

- \( \text{saturate} \) computes a fraction \( \text{oef}_i \) of \( \text{oef} \) which preserves \( h(\text{oef}, s) \) of (later: subset of) all states \( S \)
- \( \langle \text{oef}_1, \ldots, \text{oef}_n \rangle \) is a cost partitioning (CP)
- CP property: \( \forall l \in L : \sum_{i=1}^n \text{oef}_i(l) \leq \text{oef}(l) \)
- \( h_1(\text{oef}_1, s) + \ldots + h_n(\text{oef}_n, s) \) is admissible

Generalizations of SCP

| \( h(\text{oef}_1, s) \) is goal distance estimate of state \( s \) in \( S \)
| \( h \) is admissible if \( h(\text{oef}_1, s) \leq h^*(\text{oef}_1, s) \) for all states \( s \) and \( h^* \) is perfect estimate
| Abstraction is simpler version of task where a partitioning of the states \( S \) defines the abstract states
| Abstraction heuristic maps states to goal distance of corresponding abstract state in the abstraction
| Abstraction heuristics are admissible

Experiments

<table>
<thead>
<tr>
<th>( \text{generalization (1)} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \text{generalization (2)} )</td>
</tr>
<tr>
<td>operators (a)</td>
</tr>
<tr>
<td>transitions (c)</td>
</tr>
</tbody>
</table>

| \( h(\text{tcf}_1, s) \) (as alternative to \( \text{oef}_i \))
| Heuristic estimate in unsolvable state is \( \infty \) independent of \( \text{tcf}_1 \)
| Almost no value in cost assignment \( \neq 0 \)

Our Contributions

- Unify (1) and (2)
- Initial costs \( \text{tcf}(l) = 1 \) for all \( l \in L \)
- Edge and node denotations (b),(c),(d)
- (b) and (d) saturate for reachable states
- \( h(s_0) : h(b) = h(c) = 2 + 0 < 2 + 1 = h(d) \)
- Fast computation of \( h(\text{tcf}_1, s) \)
- Backward search in abstraction avoiding abstract weight computations
- Make use of lower bound 0 because \( \text{tcf} \) is always nonnegative
- Restrictions on \( \text{tcf}_1 \) (as alternative to \( \text{oef}_i \))

Induced Transition System

A Planning task \( \Pi \) induces a weighted transition system \( T = (S, T, s_0, s_1, \ldots, l_n, \text{oef}) \) with

- \( S \): set of states, \( L \): set of operator labels,
- \( T \): set of transitions \( T \subseteq S \times L \times S \),
- \( s_0 \in S \): initial state, \( S \subseteq S \) set of goal states,
- \( \text{oef} : L \rightarrow \mathbb{R} \) the operator costs (nonnegative) Opt. solution for \( \Pi \) correspond to path \( (s_0, l_1, s_1, \ldots, l_n, s_n) \) in \( T \) where \( s_n \in S \), with cheapest cost \( \sum_{i=1}^n \text{oef}(l_i) \).

Abstractions and Heuristics

| \hline
| \( h(\text{oef}, s) \) is the cost \( \sum_{i=1}^n \text{oef}(l_i) \)
| \( h \) is admissible if \( h(\text{oef}, s) \leq h^*(\text{oef}, s) \) for all states \( s \) and \( h^* \) is perfect estimate
| \hline Abstraction is simpler version of task where a partitioning of the states \( S \) defines the abstract states
| \hline Abstraction heuristic maps states to goal distance of corresponding abstract state in the abstraction
| \hline Abstraction heuristics are admissible

| \hline
| \( h(\text{oef}_1, s) \) is a cost partitioning \( \langle \text{oef}_1, \ldots, \text{oef}_n \rangle \)
| \hline CP property: \( \forall l \in L : \sum_{i=1}^n \text{oef}(l) \leq \text{oef}(l) \)
| \hline \( h_1(\text{oef}_1, s) + \ldots + h_n(\text{oef}_n, s) \) is admissible
| \hline

Experiments

<table>
<thead>
<tr>
<th>( \text{generalization (1)} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \text{generalization (2)} )</td>
</tr>
<tr>
<td>operators (a)</td>
</tr>
<tr>
<td>transitions (c)</td>
</tr>
</tbody>
</table>

| \hline
| \( h(\text{tcf}_1, s) \) (as alternative to \( \text{oef}_1 \))
| Heuristic estimate in unsolvable state is \( \infty \) independent of \( \text{tcf}_1 \)
| Almost no value in cost assignment \( \neq 0 \)
| \hline

Experiments

| \hline
| \( h(\text{tcf}_1, s) \) (as alternative to \( \text{oef}_1 \))
| Heuristic estimate in unsolvable state is \( \infty \) independent of \( \text{tcf}_1 \)
| Almost no value in cost assignment \( \neq 0 \)

| \hline
| \( h(\text{tcf}_1, s) \) (as alternative to \( \text{oef}_1 \))
| Heuristic estimate in unsolvable state is \( \infty \) independent of \( \text{tcf}_1 \)
| Almost no value in cost assignment \( \neq 0 \)

| \hline
| \( h(\text{tcf}_1, s) \) (as alternative to \( \text{oef}_1 \))
| Heuristic estimate in unsolvable state is \( \infty \) independent of \( \text{tcf}_1 \)
| Almost no value in cost assignment \( \neq 0 \)
| \hline

Experiments

| \hline
| \( h(\text{tcf}_1, s) \) (as alternative to \( \text{oef}_1 \))
| Heuristic estimate in unsolvable state is \( \infty \) independent of \( \text{tcf}_1 \)
| Almost no value in cost assignment \( \neq 0 \)

| \hline
| \( h(\text{tcf}_1, s) \) (as alternative to \( \text{oef}_1 \))
| Heuristic estimate in unsolvable state is \( \infty \) independent of \( \text{tcf}_1 \)
| Almost no value in cost assignment \( \neq 0 \)

| \hline
| \( h(\text{tcf}_1, s) \) (as alternative to \( \text{oef}_1 \))
| Heuristic estimate in unsolvable state is \( \infty \) independent of \( \text{tcf}_1 \)
| Almost no value in cost assignment \( \neq 0 \)

| \hline
| \( h(\text{tcf}_1, s) \) (as alternative to \( \text{oef}_1 \))
| Heuristic estimate in unsolvable state is \( \infty \) independent of \( \text{tcf}_1 \)
| Almost no value in cost assignment \( \neq 0 \)