Mini-buckets

Rina Dechter
275b

Mini-Bucket Heuristics for Improved Search
Kask and Dechter

Mini-buckets: “local inference”
- The idea is similar to i-consistency: bound the size of recorded dependencies
- Computation in a bucket is time and space exponential in the number of variables involved
- Therefore, partition functions in a bucket into “mini-buckets” on smaller number of variables

Mini-bucket approximation: MPE task
Split a bucket into mini-buckets => bound complexity

Approx-mpe(i)
Input: i - max number of variables allowed in a mini-bucket
Output: (lower bound (P of a sub-optimal solution), upper bound)

Properties of approx-mpe(i)
- Complexity: $O(\exp(2i))$ time and $O(\exp(i))$ time.
- Accuracy: determined by upper/lower (U/L) bound.
- As $i$ increases, both accuracy and complexity increase.
- Possible use of mini-bucket approximations:
  - As anytime algorithms (Dechter and Rish, 1997)
  - As heuristics in best-first search (Kask and Dechter, 1999)
- Other tasks: similar mini-bucket approximations for: belief updating, MAP and MEU (Dechter and Rish, 1997)

Anytime Approximation
 anytime-mpe(c)
 Initialize: $i = i_0$
 While time and space resources are available
   $i \leftarrow i + 1$  upper bound computed by approx - mpe(i)
   $L \leftarrow$ lower bound computed by approx - mpe(i)
   keep the best solution found so far
   if $1 \leq \frac{L}{U} \leq 1 + \epsilon$, return solution
 end
 return the largest $L$ and the smallest $U$
**Heuristic search**

- Mini-buckets record upper-bound heuristics
- The evaluation function over $T_p = (x_1, \ldots, x_p)$
  
  \[
  f(T_p) = g(T_p)h(T_p)
  \]
  
  \[
  g(T_p) = \prod_{i=1}^{p} P(x_i | p_{a_i})
  \]
  
  \[
  h(T_p) = \prod_{i=1}^{p} h_{p_i}
  \]

- **Best-first:** expand a node with maximal evaluation function
- **Branch and Bound:** prune if $f \geq$ upper bound
- **Properties:**
  - an exact algorithm
  - Better heuristics lead to more pruning

---

**Heuristic Function**

Given a cost function $P(a,b,c,d,e) = P(a) \cdot P(b|a) \cdot P(c|a) \cdot P(e|b,c) \cdot P(d|b,a)$

Define an evaluation function over a partial assignment as the probability of its best extension

\[
 f^*(a,e,d) = \max_{b,c} P(a,b,c,d,e) = P(a) \cdot \max_{b,c} P(b|a) \cdot P(c|a) \cdot P(e|b,c) \cdot P(d|b,a)
\]

\[
 f(a,e,d) = g(a,e,d) \cdot H^*(a,e,d)
\]

The heuristic function $H$ is compiled during the preprocessing stage of the Mini-Bucket algorithm.

---

**Properties**

- Heuristic is monotone
- Heuristic is admissible
- Heuristic is computed in linear time

**IMPORTANT:**

- Mini-buckets generate heuristics of varying strength using control parameter $I$
- Higher bound $I$ -> more preprocessing -> stronger heuristics -> less search
- Allows controlled trade-off between preprocessing and search

---

**Empirical Evaluation of mini-bucket heuristics**

Random Coding, $K=100$, noise 0.32