Parallel Beam Search Algorithms for Domain-Independent Dynamic Programming

Ryo Kuroiwa and J. Christopher Beck
Department of Mechanical and Industrial Engineering, University of Toronto

Domain-Independent Dynamic Programming (DIDP)

Combinatorial Optimization Problem

DP Model

General-Purpose DP Solver

DIDPy: Python Modeling Interface for DIDP

\[
V(u, i) = \min_{j \in U} c_{ij} + V(u \setminus \{j\}, i)
\]

Domain-Independent Dynamic Programming (DIDP)

- Compute \(V(\emptyset, 0)\), \(V(\emptyset, i) = c_{0i}\)
- State Transition Graph for a DP Model

Parallel Beam Search for DIDP

Heuristic Search for DIDP

- \(f\)-value: \(g + h\)
- \(g\)-value: actual path cost
- \(h\)-value: estimation by a heuristic function (given with a model in current DIDP)
- \(f\)-value: priority to expand

SOTA DIDP Solver: CABS

[Kuroiwa and Beck 2023c]

Hash Distributed Beam Search (HDBS)

Hash Distributed Beam Search (HDBS)

Heuristic Search for DIDP

- Beam search expands the best \(b\) states minimizing the \(f\)-value in each layer
- CABS [Zhang 1998] repeats beam search with increasing \(b\) until proving optimality
- Adaptation of HDA* [Kishimoto+ 2013]

- Send a state to a thread assigned by its hash value with message passing
- Each thread expands \(b\) states

Layer Synchronization

HDBS1: Immediate Layer Synchronization

- Expand the best states in layer \(i\)
- Send and receive successors in layer \(i + 1\)
- Synchronize all threads
- Go to the next layer (\(i \leftarrow i + 1\))

HDBS2: Delayed Layer Synchronization

- Expand the best states in layer \(i\)
- Check if all threads finish layer \(i - 1\)
- Send and receive successors in layer \(i + 1\)
- Notify other threads that layer \(i\) is finished
- Go to the next layer (\(i \leftarrow i + 1\))

Experimental Evaluation

- 5-min, 188GB
- SBS: parallel beam search for DIDP with a concurrent hash table
- Gurobi: mixed-integer programming (MIP)
- CPO: CP Optimizer, constraint programming

<table>
<thead>
<tr>
<th>Problem</th>
<th>Description</th>
<th>Gurobi</th>
<th>CPO</th>
<th>DIDP (HDBS2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>TSPTW (340)</td>
<td>TSP with time</td>
<td>239/4.2</td>
<td>27/0.1</td>
<td>262/13.3</td>
</tr>
<tr>
<td>CVRP (207)</td>
<td>vehicle routing</td>
<td>29/5.3</td>
<td>0/-</td>
<td>8/ 9.3</td>
</tr>
<tr>
<td>SALBP-1 (2100)</td>
<td>line balancing</td>
<td>1351/1.3</td>
<td>1581/1.4</td>
<td>1826/18.8</td>
</tr>
<tr>
<td>Bin Packing (1615)</td>
<td>bin packing</td>
<td>1192/6.4</td>
<td>1251/9.2</td>
<td>1239/39.6</td>
</tr>
<tr>
<td>MOSP (570)</td>
<td>manufacturing</td>
<td>238/3.1</td>
<td>397/0.3</td>
<td>531/ 9.0</td>
</tr>
<tr>
<td>Graph-Clear (135)</td>
<td>building security</td>
<td>162/2.0</td>
<td>4/3.2</td>
<td>113/10.3</td>
</tr>
</tbody>
</table>

- #optimally solved / mean speedup (32 threads)