Cost-Optimal Planning with Complementary Pattern-Databases

Santiago Franco

Huddersfield University

A Tale of 4 planners

1. RIDA. Full Sampling(CCs). ICAPS 14
2. GHS. Stratified Sampling. IJCAI 16.
3. CPC. Dynamic Parameters. IJCAI 17.
4. Complementary. Learning Parameters. IPC 18.
5. Future: Learning Choices.

Choices

1. Cost-Partition
2. Representation
3. Selection
4. Pattern Generators
5. Seeders
6. Learning
7. Generation Rules
8. Local Search
9. Combinations

Existing Methods

Algorithm	iPDB	gaPDB	Gamer	CPC
Evaluation	Random walk	Avg-h	Avg-h	SS
Candidate	HC/HC-VNS	GA	HC	GA'
PDB Size Limit	Fixed	Fixed	—	Dynamic
Aggregation	h^C	$0/1P$	—	$0/1P + h^C$

Table 6: Comparison of PC generation algorithms.

1. Cost Partition

- Ask Florian and Jendrik, that is why they are here :-)
- We use the humble Zero-One partition
- We do not claim this to be the best approach, but it works well enough.
- Saturated Cost Partitioning probably better
- But there is a trade-off between combination system calculations vs generating more patterns.

2. Symbolic Representation

Symbolic:

1. **No a priori size limit (cross product)**
2. **Larger search spaces than explicit.**
3. **More expensive to initialize and query than explicit.**
4. **Compression level is unknown a priori!**

2. Explicit Representation

1) Fast to initialize and query

2) For small PDBs, it can be faster to generate.

3) Great in “disjunctive-friendly” domains like Visitall.

4) Scorpion showed they are still very competitive.

3. Selection Metric

Metric:

- Time Metric
- Size Metric
- Avg Heuristic Value

Sampling Technique:

- Full Sample using Culprit Counters(RIDA)
- Random Walk (iPDB Sampling)
- Stratified Sampling

4. Sampling Algorithms

- **Random walk with variable depth (iPDB) AAAI 07**
- **Full compressed lossless sample using Culprit Counters (RIDA) ICAPS 14**
- **Stratified Sampling with Culprit Counters (Levi + Franco) IJCAI 16**

5. Pattern Generators

- **RBP: GA-style (FD's implementation of Edelkamp GA), Prune unconnected a posteriori.**
- **CBP: Only Connected, Disjoint, Asymteric.**
- **Gamer:Single pattern, iterative, avg h value.**
- **Perimeters:All variables, partial PDB growth.**

6. Seeders

- **Explored Options:**
  - Perimeter: Arbitrary large time and memory limit.
  - Gamer: The same limits as Perimeter.
  - LmCut: Sampling issue, asymetric computational costs.
- **PDBs are generated and selected to complement Seeders.**
- **Smart enough to ignore Seeder if not competitive.**

7. Learning

- **Pattern Generator**
- **PDB sizes (crossproduct)**
- **Max number of variables**
- **Number of patterns (GAs)**
- **Time per PDB (partial PDBs, Complementary)**
- **Disjoint or not**
- **Number of starting goal variables**
- **New, improve or local search for pattern generation.**

8. Local Search

- **Fully Random: Mutations**
- **Deterministic: Gamer-like improvements**
- **Merge smaller patterns**

9. Combining PDBs

- **Cannonical: Expensive but safe**
- **Sample-based: Greedy, Faster TPN, but potentially negative.**
- **Clear connection with cost partititioning and admissible heuristic combination theory.**

10. PDB generation types

1. Fully
2. Incremental
3. On-line
4. Hierarchical (Holte)

IPC Complementary1 Algorithm

Algorithm 1 Complementary PDBs Creation

Require: base heuristic h_{base}, time and memory limits t and m respectively UCB1 PDB size range S_{min}, S_{max}.
Ensure: Selected set of pattern collections $\mathcal{P}_{sel}$

1: $\mathcal{P}_{sel} = \emptyset$ // *$\mathcal{P}_{sel}$ is a set of pattern collections*
2: $\mathcal{P} = \emptyset$ // *$\mathcal{P}$ is a pattern collection*
3: Ch // Ch UCB1 choices: new generating $\mathcal{P}$ parameters or grow un-terminated $\mathcal{P} \in \mathcal{P}_{sel}$ or local search using $\mathcal{P} \in \mathcal{P}_{sel}$
4: $Seed \leftarrow$ search heuristic or pattern(s) with arbitrary time/size limits or $\emptyset$.
5: $SampleSearchSpace$ is a time-limited sample, given a $\mathcal{P}_{sel}$ using either strat. sampling or an iPDB-style Random walk.
6: EM // EM is the evaluation method given a candidate $\mathcal{P}$, a $SampleSearchSpace$ and a metric(time or size). True if improvement found.
7: $SelPDBs \leftarrow Seed$
8: $SampleSearchSpace(\mathcal{P}_{sel})$
9: **while** time t or memory m limits are not exceeded **do**
10: $T_{cost} = StartTimer$
11: Choose a representation(symbolic or explicit)
12: $Ch \leftarrow GenerateUCB1Choices$
13: $\mathcal{P} \leftarrow Execute(Ch)$
14: **if** $EM(\mathcal{P}_{sel}, \mathcal{P}, SampleSearchSpace)$ **then**
15: $\mathcal{P}_{sel} \leftarrow \mathcal{P}$
16: $UpdateRewards(T_{cost})$
17: $SampleSearchSpace(\mathcal{P}_{sel})$
18: $ClearDom(\mathcal{P} \in \mathcal{P}_{sel}, SampleSearchSpace)$
19: $UpdateCosts(T_{cost})$
20: **return** Cannonical($\mathcal{P}_{sel}$)

Future Work

1) Online PDBs (Holte, Haslum)

2) Reformulation of SAS+ representations (Similar to symbolic search)

3) Advanced cost partitioning with symbolic search

4) Learning beyond parameters: Terminate, local search or new patterns?

5) New pattern generator methods, e.g. bin packing.

Results: IPC 18 Complementary1

Table 1: Coverage of Complementary1 Modules. Reg stands for all components active. "NoPer" stands for perimeter PDB inactivated. "+BinPack" stands for using PDBs generated by bin packing generator. "CBP", "Cgamer" and "BinPackOnly" rows also have Perimeter inactive.

Domain/Method	Agr	Cal	DN	Nur	OSS	PNA	Set	Sna	Spi	Ter	Total
Comp1/Reg	10	11	13	12	**12**	**19**	**9**	10	**12**	**16**	124
Comp1/NoPer+BinPack	10	**12**	**14**	**14**	**12**	6	**9**	12	11	**16**	126
Comp1/NoBinPack	6	11	13	**14**	**12**	**19**	**9**	11	11	**16**	122
Comp1/CBP+BinPack	8	**12**	**14**	13	**12**	**18**	**9**	11	11	**16**	124
Comp1/CBP-NoBinPack	6	**12**	**14**	13	**12**	18	**9**	9	11	**16**	120
Comp1/Gamer+BinPack	**13**	**12**	**14**	**14**	**12**	17	**9**	12	11	**16**	**130**
Comp1/Gamer-BinPack	8	12	12	**16**	**12**	18	**9**	**14**	11	**16**	128
Comp/BinPackOnly	7	**12**	14	12	**12**	7	**9**	11	11	12	107
Solved by any of the Comp1 methods above											
*	14	12	14	16	12	20	9	14	12	16	139
Competition result below included for Completeness											
Comp1	10	11	14	**13**	13	17	8	11	**11**	**16**	124

Results: Symbolic Vs Explicit, IJCAI 17

	RBP	CBP	50/50	UCB1
Explicit	908	864	923	**936**
Symbolic	973	909	1,000	**1,009**

Table 1: Planner's coverage while using different bin packing approaches.

Results: Fixed vs Dynamic PDB Size

	Adjustable	10^4	10^5	10^6	10^7	10^8
Explicit	**936**	891	923	930	918	316
Symbolic	**1,009**	884	910	919	924	940

Table 2: Planner's coverage with our scheme to automatically adjust the PDB size limit and with different fixed values of PDB size limit.

Additional Results IJCAI 17

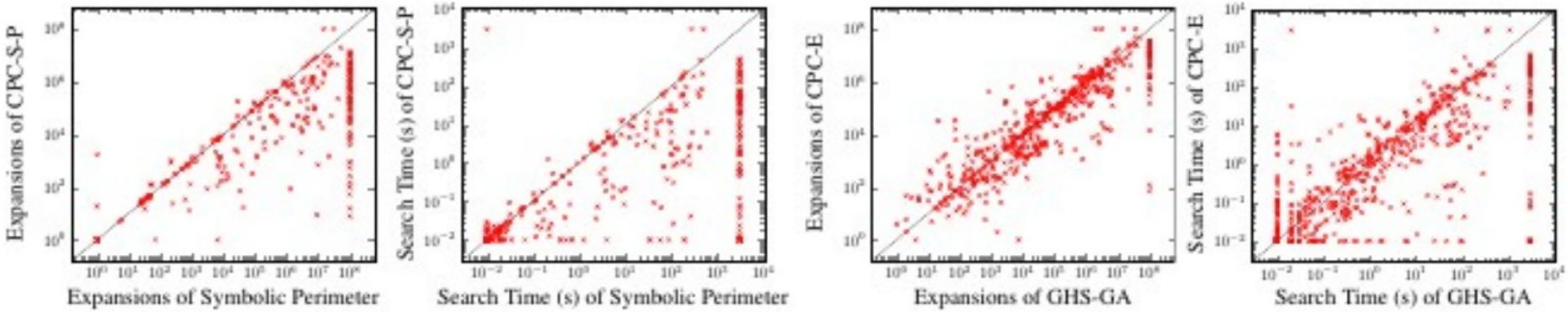

Figure 1: Search time and expanded nodes until last f-layer for CPC-E vs. GHS-GA and CPC-S-P vs. P.

Explicit			Symbolic		
AVG	RS	SS	AVG	RS	SS
872	891	**936**	895	**1,009**	**1,009**

Table 4: Coverage of alternative sampling methods.

CPC-S	P	max(P, CPC-S)	CPC-S-P
1,009	962	1,014	**1,055**

Table 5: Coverage of CPC complementing a perimeter heuristic vs. maximum of regular CPC and the same heuristic.

Further Reading Authors

Cost partitioning: Pommering, Seipp.

Sampling: Levi, Barley, Helmert, Franco.

Pattern Generation: Edelkamp, Haslum, Kissman, Torralba,Franco.

Symbolic Search: Torralba, Edelkamp, Haslum.

Local Search: Edelkamp,Franco.