# Automated Pattern Selection using MiniZinc

S. Edelkamp, M. Martinez, I. Moraru

King's College of London

{*firstname.lastname*}*@kcl.ac.uk*

September 6, 2018

# Overview

# AI Planning

**Planning** is the discipline that has the task of coming up with a *sequence of actions* that will achieve a goal.

# AI Planning Task

**Definition (Planning Task)**

A planning problem defined in propositional STRIPS is a 4-tuple $\Pi = \langle V, I, G, A \rangle$ where:

- V is a finite set of state variables;
- $I \subseteq V$ is the initial state of the problem;
- $G \subseteq V$ a set of goal variables;
- $A$ is a finite set of actions (each action being composed out of preconditions, add effects, delete effects and a cost[a]).

[a] Action costs are an extension of traditional propositional STRIPS

# Heuristic Search

**Heuristics** are meant to be an estimate of the remaining distance from a node to the goal.

- Relaxations of the problem constraints that solve a relaxed problem exactly;
- We can do an *informed* search through the state space and access one state that is *more promising* than the rest by using this estimate.

**Definition (Heuristic)**

*Heuristic h* is a function that maps states, $\mathcal{S}$, to positive numbers:

$$h : \mathcal{S} \rightarrow \mathbf{R}_{+} \cup \{0, \infty\}$$

# Pattern Databases

**Pattern databases (PDBs)** map the state space of a planning task onto an abstract state space by considering only a subset of the planning variables.

- The pre-computed optimal distance from each abstract state to an abstract goal state is stored in a lookup table.
- Combining multiple PDBs can result in better heuristic estimates, than what would be returned by using only one.

We use the approach developed by Haslum et al. (2007), in which it takes the maximum estimate over all additive PDB subsets.

# Pattern Databases - Limitations

The main *limitation* of PDBs is the amount of memory needed, as during construction the abstract state space may prove to be too large;

- To deal with memory requirements, PDBs have been extended to symbolic search, which represents state sets in the search compactly as decision diagrams (Edelkamp, 2002);

# Pattern Databases - Limitations

The main *limitation* of PDBs is the amount of memory needed, as during construction the abstract state space may prove to be too large;

- To deal with memory requirements, PDBs have been extended to symbolic search, which represents state sets in the search compactly as decision diagrams (Edelkamp, 2002);

Another issue is the **automated selection** of the most informative patterns, which remains a combinatorial challenge.

# Pattern Selection

We identify a pattern database with its abstraction function $\phi$.

- The *fitness* $f$ of a pattern database $\phi$ (represented as a set of pairs $(a, h(a)) \in \phi$) is the average heuristic estimate:

$$f(\phi) = \sum_{(a,h(a))\in\phi} h(a)/|\phi|$$

- For several PDBs and cost partitioning $\gamma$, the values are added linearly:

$$f(\phi_1, \ldots, \phi_l) = \sum_{i=1}^{l} \sum_{(a,h_{i,\gamma}(a))\in\phi_i} h_{i,\gamma}(a)/|\phi_i|$$

The **pattern selection problem** is to find a selection of pattern databases that fit into main memory, and optimises $f$.

# Bin Packing Problem

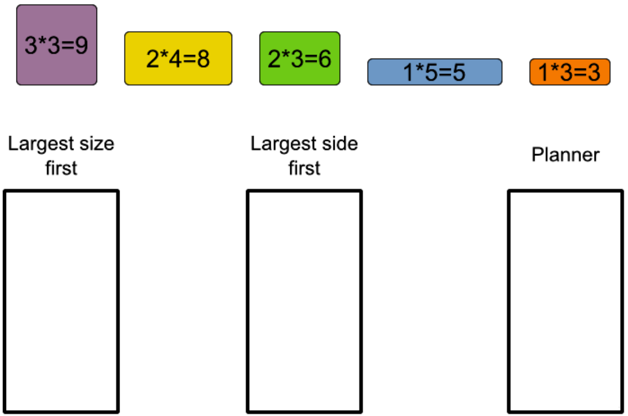
The **Bin Packing Problem (BPP)** is one of the first problems shown to be NP-hard (Garey and Johnson, 1979).

## Definition (Bin Packing)

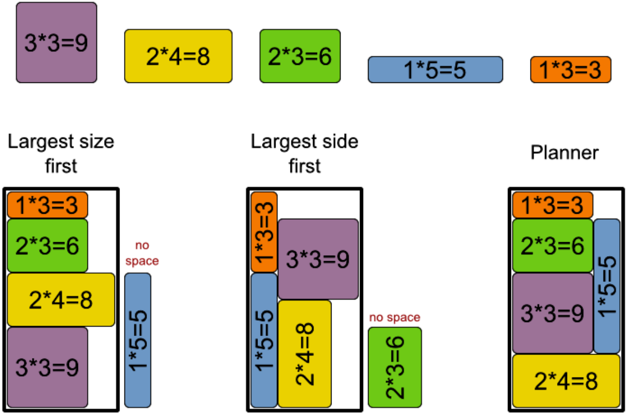
Given objects of integer size $a_1, \ldots, a_n$ and maximum bin size $C$, the problem is to find the minimum number of bins $k$ so that the established mapping $f : \{1, \ldots, n\} \rightarrow \{1, \ldots, k\}$ of objects to bins maintains

$$\sum_{f(a)=i} a \leq C \text{ for all } i \leq k$$

# Bin Packing Problem - Example (1/2)



# Bin Packing Problem - Example (2/2)



# MiniZinc for Pattern Selection

Our intent for solving the BPP is to be solver-independent, and to apply descriptive models:

- We looked methods that come in the form of a framework, and have choosen to use MiniZinc a free and open-source constraint modeling language;
- MiniZinc is compiled into the input of a range of different constraint-based solvers including Gurobi and Gecode.

# MiniZinc Model 1/2

## Example (Variables for BPP)

```
int: num_bins;
int: num_objects;
array[1..num_objects] of int: object;
int: bin_capacity;
array[1..num_bins, 1..num_objects] of var 0..1: bins;
array[1..num_bins] of var 0..bin_capacity: bins_of_load;
var 0..num_bins: num_loaded_bins;
```

# MiniZinc Model 2/2

**Example (Constraints for BPP)**

```
forall(b in 1..num_bins)
 (bin_loads[b] = sum(s in 1..num_objects)
 (object[s]*bins[b,s])) /\
sum(s in 1..num_objects) (object[s]) =
 sum(b in 1..num_bins) (bin_loads[b]) /\
forall(s in 1..num_objects)
 (sum(b in 1..num_bins)(bins[b,s]) = 1) /\
decreasing(bin_loads)  /\ bin_loads[1] > 0 /\
num_loaded_bins = sum(b in 1..num_bins)
 (if bins_of_load[b] > 0 then 1 else 0 endif);
```

# Experiments Results

In this table we compare the results from the IPC-18, optimal track, of planners Planning-PDBs, MinizincPDB Complamentary2- the first two being an extension of the last including our work on using different bin packing strategies.

| Domain | Planning-PDBs | Complementary2 | MinizincPDB |
|---|---|---|---|
| **Nurikabe** | 11 | **12** | **12** |
| **Settlers** | 8 | **9** | 8 |
| **DataNet** | **14** | 12 | 13 |
| **Caldera** | **12** | **12** | **12** |
| **Snake** | 13 | **14** | 11 |
| **Spider** | 11 | **12** | 11 |
| **Total** | 69 | **71** | 67 |

The numbers represent the amount of solved problems on each domain out of a total of 20 problems.

# MiniZinc for Pattern Selection

- Our first attempt to integrating MiniZinc library version 1.6 resulted to a number of problems failing - as such we did not include it in the IPC submission;
- We have since tried to integrate version 2.2, receiving results but hiting some interesing errors on the way.

# Future Development

- **Short term goal** - integrate MiniZinc as it shows potetial to give good results for more components of our planner;
- **Medium term goal** - analyze how different variable orderings affect the pattern selection process;
- **Long term goal** - make an automated Pattern Selection method that will not use a GA (Edelkamp, 2007).

Thank you very much!