Bin Packing strategies for Pattern Selection

Moisés Martínez, Ionut Moraru and Stefan Edelkamp

King's College London

moises.martinez@kcl.ac.uk

October 4, 2018

Overview

1. Motivation
   - PDBs
2. Automated Pattern Selection
   - Bin Packing Strategies
3. Experiments
   - Results

Motivation

- Cost-optimal planning: (V, O, I, G)
- Improve the pattern generation process for PDBs
- Pattern generation methods
  - **Evolutionary Algorithm**
  - Hill-climbing algorithm

Pattern Databases (PDBs)

Idea

pre-compute and store the cost of the solution for all possible sub-problems (abstract problems) in a database.

Pattern Databases (PDBs)

Idea

pre-compute and store the cost of the solution for all possible sub-problems (abstract problems) in a database.

Limitations

Build a PDB is an expensive operation in **time** and **memory**, but at least just needs to be computed only one time before search.

Pattern Databases (PDBs)

Idea

pre-compute and store the cost of the solution for all possible sub-problems (abstract problems) in a database.

Limitations

Build a PDB is an expensive operation in **time** and **memory**, but at least just needs to be computed only one time before search.

Question

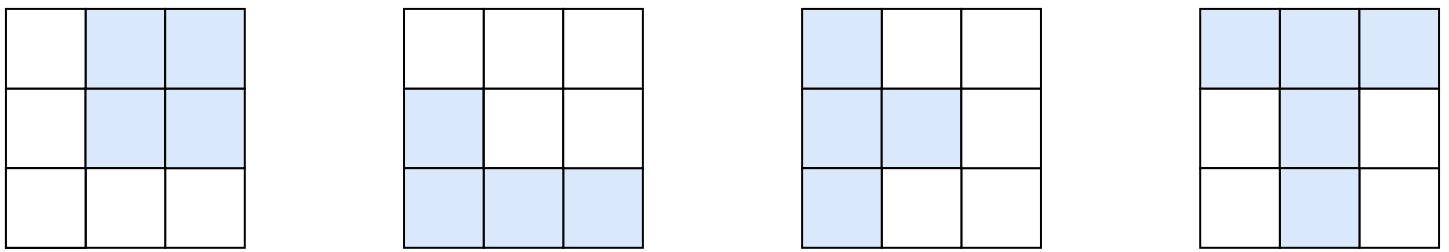
How to create PDBs on the best way?

Pattern Selection

The objective is to find **the best patterns** to build the PDBs

objectives

- Maximize the quality of the heuristic
- Minimize the space and time during construction

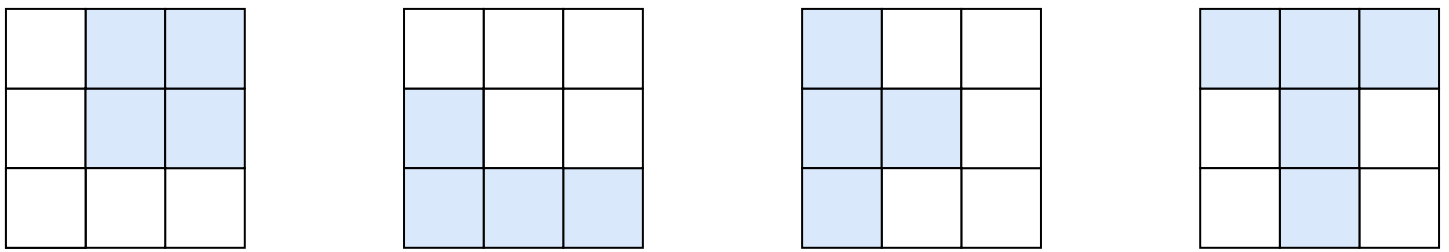


Pattern Selection

The objective is to find **the best patterns** to build the PDBs

objectives

- Maximize the quality of the heuristic
- Minimize the space and time during construction



Solution?

Pattern selection problem can be cast as an optimization problem

Pattern Selection

The most common solution consists on find good patterns by means of **local search** techniques

Approaches

Two approaches from the literature:

- Edelkamp (2007): using an evolutionary algorithm
- Haslum et al. (2007): using hill-climbing algorithm

Pattern Selection

The most common solution consists on find good patterns by means of **local search** techniques

Approaches

Two approaches from the literature:

- Edelkamp (2007): using an evolutionary algorithm
- Haslum et al. (2007): using hill-climbing algorithm

Question?

How can we improve or replace these algorithms?

Bin Packing Problem

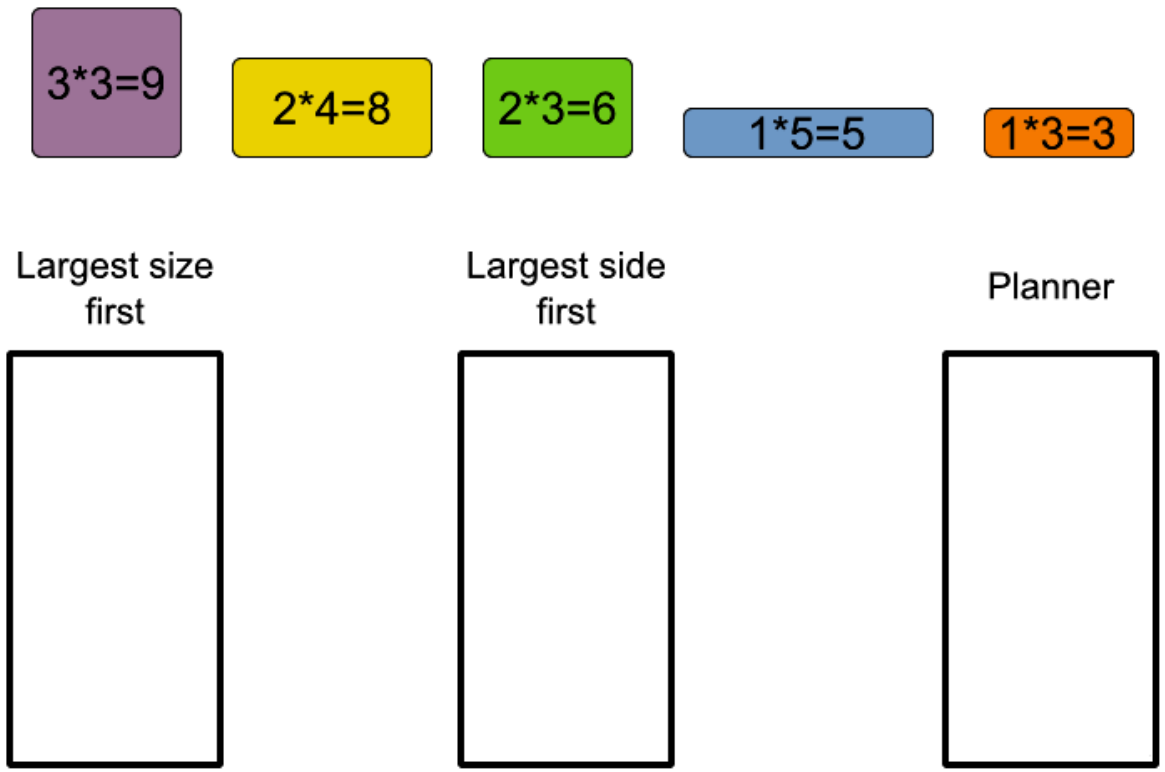
The **Bin Packing Problem** is one of the first problems shown to be NP-hard (Garey and Johnson, 1979).

Bin Packing definition

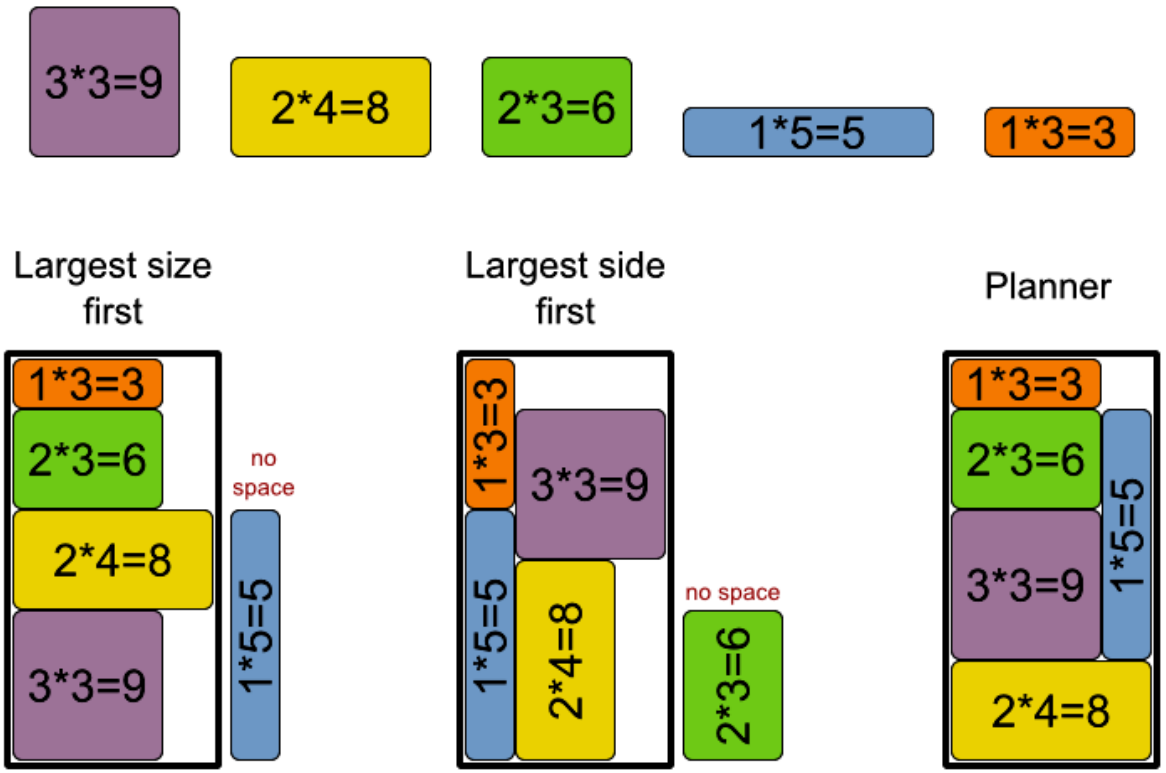
Given a set of variables V of integer size $1, \ldots, n$ and a set of bins B of integer capacity C, the problem is to find the minimum number of bins B so that the established mapping $f : \{1, \ldots, n\} \rightarrow \{1, \ldots, B\}$ of objects to bins maintains

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

Bin Packing Problem - Example (1/2)



Bin Packing Problem - Example (2/2)



Bin Packing Strategies

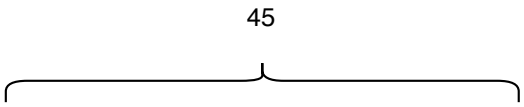
Different ways to generate the patterns using the SAS+ variables.

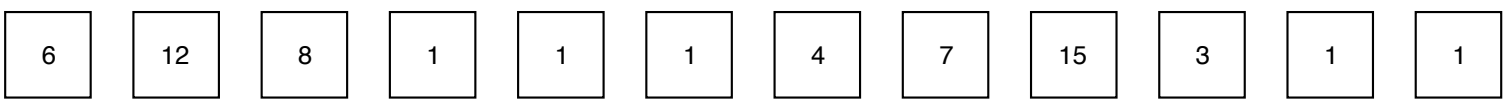
- Decreasing: Choosing bigger variables first using different strategies (sequential, random and related)
- Increasing: Choosing smaller variables first using different strategies (sequential, random and related)
- CSP approach: Modeling bin packing problem like a CSP (Constraint Satisfaction Problem).
- MCTS approach: Finding the best distribution of variable by means of random sampling of the search space.

Bin Packing definition

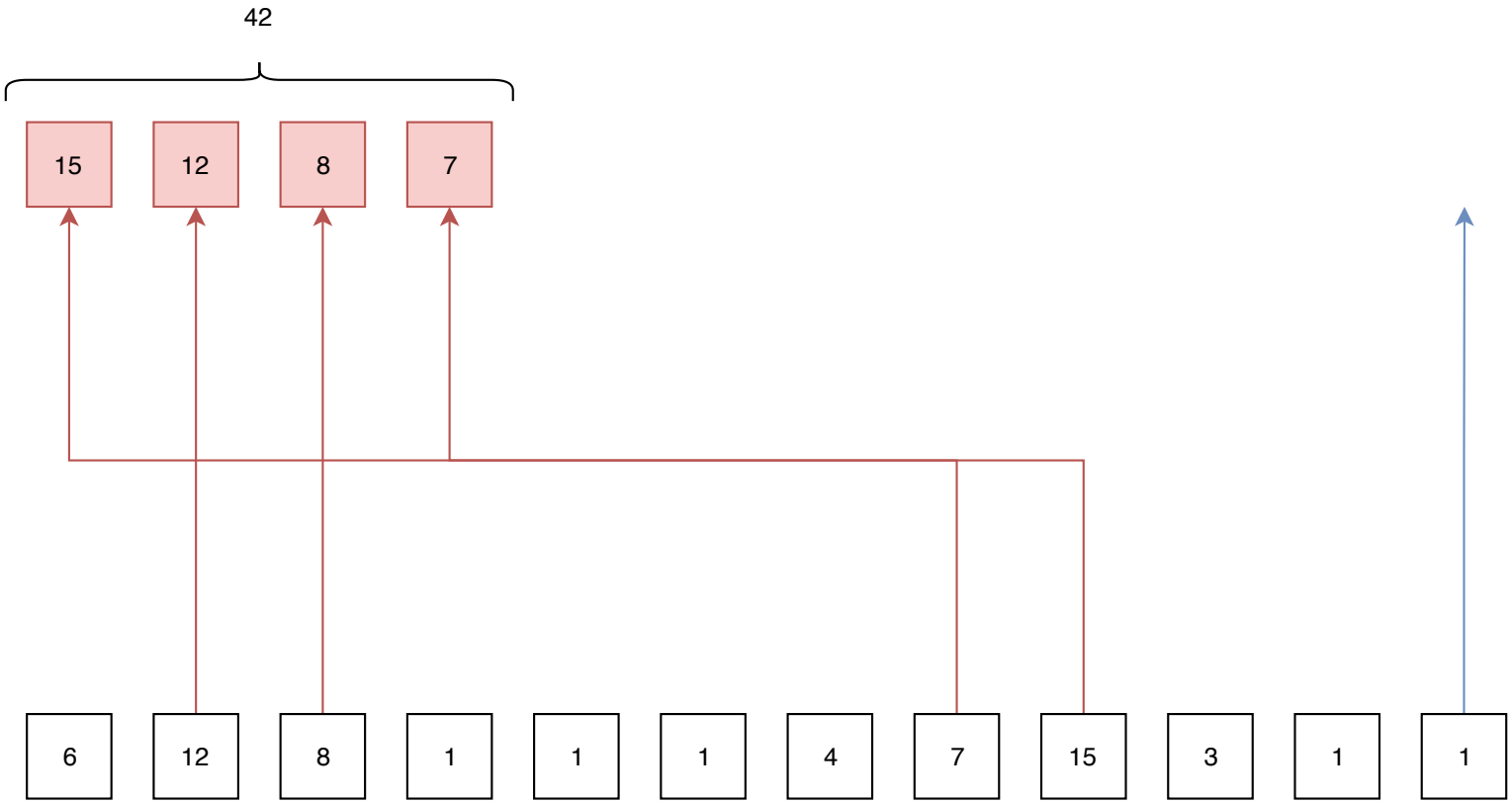
These strategies can be used in combination with evolutionary algorithm (Edelkamp 2007)

Decreasing sequential

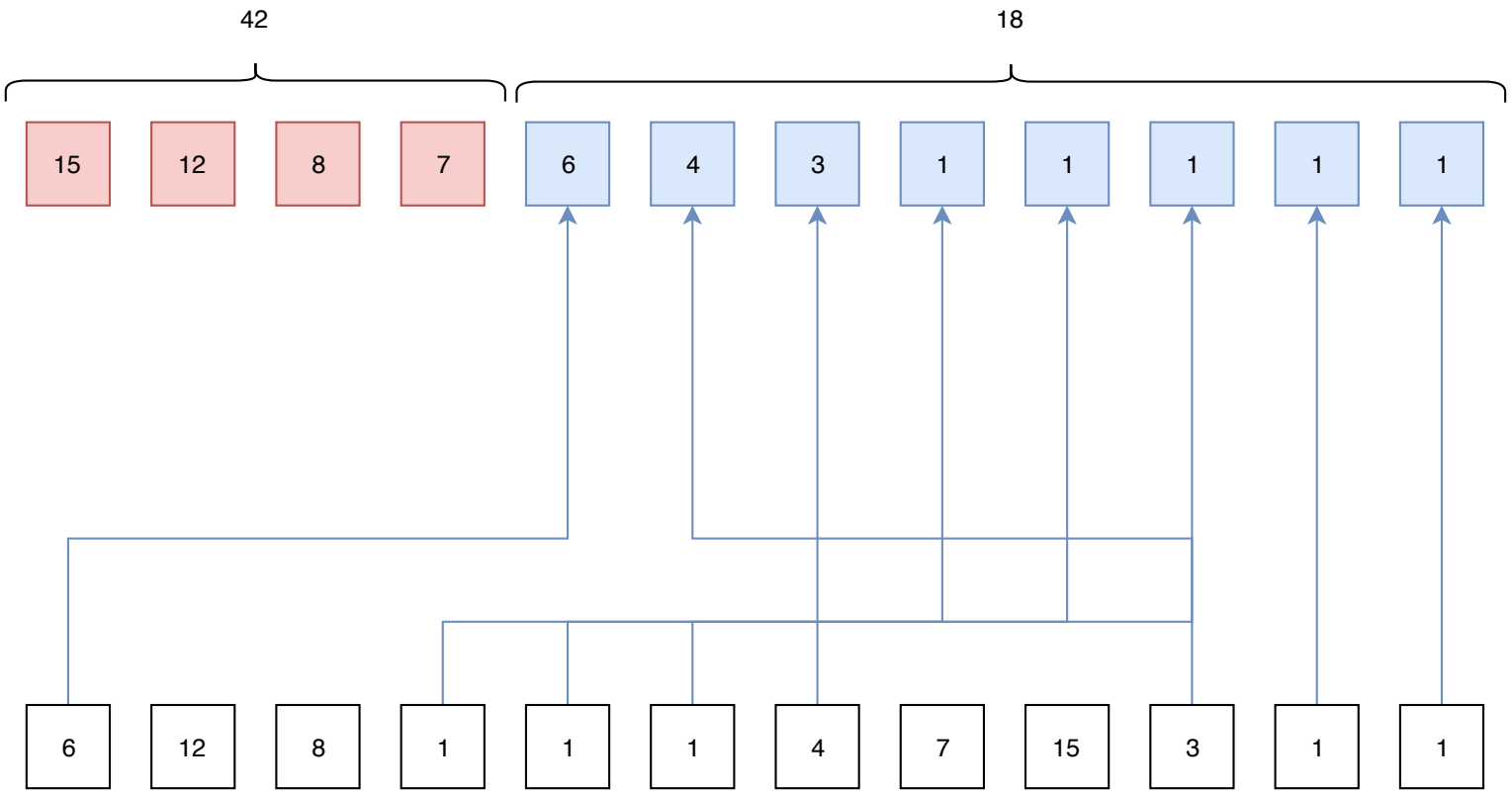




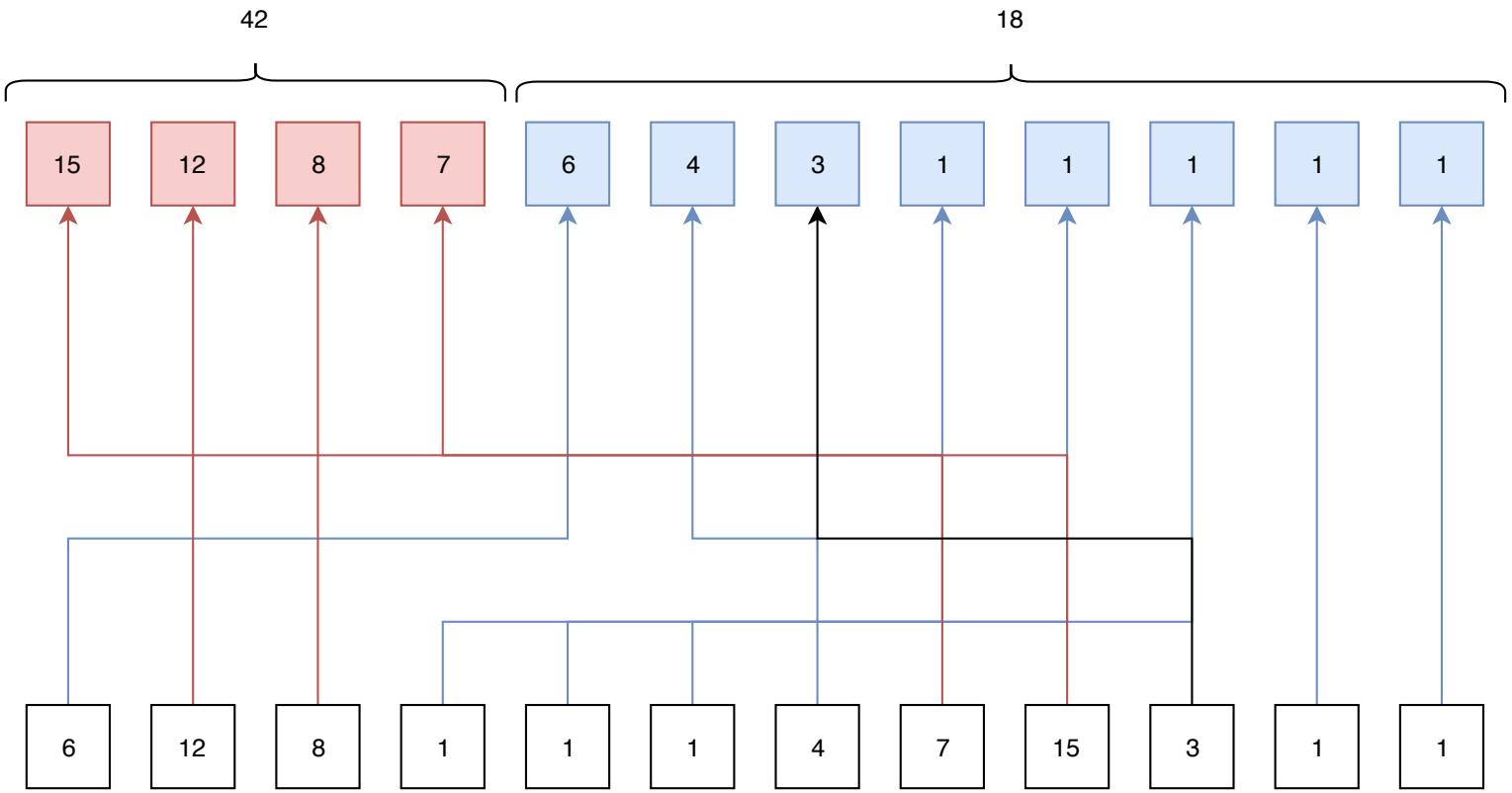
Decreasing sequential



Decreasing sequential



Decreasing sequential



Planner versions

We have combined different bin packing algorithms in order to create different strategies.

- Complementary 2: Combining two different bin packing algorithms (Increasing and decreasing sequential) with evolutionary algorithm.
- Planning-PDBs: Using Increasing related without evolutionary algorithm.
- MinizincPDB: Modeling bin packing problem as a CSP using Minizinc (Gecode solver).

Experiments Results

Domains	Agricola	Caldera	Data Network	Nurikabe	Organic Synthesis	Petri Net Alignment	Settlers	Snake	Spider	Termes	SUM
Delfi1	12	13	13	12	13	20	9	11	11	12	126
Complementary1	10	11	14	13	13	17	8	11	11	16	124
Complementary2	6	12	12	12	13	18	9	14	12	16	124
Planning-PDBs	6	12	14	11	13	18	8	13	11	16	122
Baseline: Sym-Bi	15	10	13	11	13	19	8	4	7	18	118
Scorpion	2	12	14	13	13	0	10	14	17	14	109
Delfi2	11	11	13	11	13	9	8	7	7	15	105
FDMS2	14	12	9	12	13	2	8	11	11	12	104
FDMS1	9	12	10	12	13	2	9	11	11	12	101
DecStar	0	8	14	11	14	8	8	11	13	12	99
Metis1	0	13	12	12	14	9	9	7	11	6	93
MSP	7	8	13	8	12	10	0	11	6	16	91
Metis2	0	15	12	12	14	9	0	7	12	6	87
Baseline: blind	0	8	7	11	10	7	8	12	11	10	84
MinizincPDB	-	12	13	12	-	-	8	11	11	-	67
Symple-2	1	8	9	7	13	2	0	0	5	13	58
Symple-1	0	8	9	8	13	2	0	0	4	13	57
maplan-2	2	2	9	0	6	0	0	14	1	12	46
maplan-1	0	2	12	0	6	0	0	7	10	6	43

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

Thank you very much!