

Modelling for Combinatorial Optimisation (1DL451)

Uppsala University – Autumn 2025

Report for the Project by Team **t**:

Problem

Clara CLÄVER and Whiz KIDD

4th June 2025

A Problem

... Reply ...

B Approach

... Reply ...

Our model is given in Listing 1: it has the prescribed comments as per the scope (Topics 1 to 8) of the project, it has the name `project.mzn` and the imposed structure of the provided skeleton model `project-skeleton.mzn`, and it is uploaded to Studium.

Listing 1: A MiniZinc model for ...

```
5 %% Parameters:
6 ...; % ... meaning ... (for example: X[i,j] = ... i ... j ...)
7
8 %% Derived parameters:
9 ...; % ... meaning ... (for example: X[i,j] = ... i ... j ...)
10 % None
11
12 %% Decision variables:
13 ...; % ... meaning ... (for example: X[i,j] = ... i ... j ...)
14 % Problem constraints enforced by this choice of variables:
15 % ... paraphrase ...
16 % No problem constraints are enforced by this choice of variables.
17
18 %% Redundant decision variables:
19 ...; % ... meaning ...; (non-)mutually redundant with ...
20 % None, because ...
21
22 %% Channelling constraints:
23 constraint ...; % 1-way from ... to ... / 2-way between ... and ...
24 % None, because ...
25
26 %% Problem constraints:
27 constraint ...; % ... paraphrase ...
28
```

Backend	Chuffed		CP-SAT		Gecode		Gurobi		PicatSAT		Yuck	
...	obj	time	obj	time	obj	time	obj	time	obj	time	obj	time
...

Table 1: Results for our approach to [Problem](#), which is a [minimisation / maximisation / satisfac-](#)
[tion](#) problem. In each time column: if the reported time is less than the time-out (6,000,000 mil-
liseconds here), then the [objective value in the corresponding obj column](#) was *proven* optimal;
else the timing out is indicated by `t/o` and the [objective value](#) is either the best one found
but *not* proven optimal before timing out, or ‘-’ indicating that no feasible solution was found
before timing out. Boldface indicates the [best performance](#) (time or objective value) on each
row.

```

29 %% Implied constraints:
30 constraint implied_constraint(...); % ... paraphrase ...
31 % None.
32
33 %% Symmetry-breaking constraints:
34 constraint symmetry_breaking_constraint(...); % ... paraphrase ...
35 % None.
36
37 solve
38   %% Search strategy:
39   % ... paraphrase ...:
40   :: ..._search(...)
41   %% Objective and objective function:
42   % ... paraphrase ...:
43   minimize ... | maximize ... | satisfy;
44
45 %% Pretty-print solutions:
46 output [...];

```

Symmetries. ... [problem symmetry](#) ... [model symmetry](#) ... [value / variable / index / row](#)
[/ column symmetry](#) ... [full / partial symmetry](#) ...

Efficiency. ... [implied constraints](#) ... [symmetry-breaking constraints](#) ... [reasoning annota-](#)
[tions](#) ... [search annotations](#) ...

Checklist. ... [Reply](#) (and we understand that we may lose points if there are such model
[features that we did not detect and discuss](#)) ...

Correctness. ... [Reply](#) ...

C Evaluation

All experiments were run under Linux Ubuntu 22.04.5 (64 bit) on an Intel Xeon E5520 of 2.27 GHz, with 4 processors of 4 cores each, with a 70 GiB RAM and an 8 MiB L3 cache (a ThinLinc computer of the IT department).

Table 1 gives our results. The time-out was [6,000,000](#) milliseconds.

Which backends win overall, and how do you draw that conclusion? ... [Reply](#) ...

How do the backends scale, and how do you draw that conclusion? ... [Reply](#) ...

Does the difficulty of instances monotonically increase with their size, and how do you draw that conclusion? ... [Reply](#) ...

How suitable is local search compared to systematic search, and how do you draw that conclusion? ... [Reply](#) ...

Are there any contradictions between the results? ... [Reply](#) ...

Are there any occurrences of ‘ERR’ within the results generated by the experiment script? ... [Reply](#) ...

Feedback to the Teachers

... [Reply](#) ...

Error Report

... [Reply](#) ...