Modelling for Combinatorial Optimisation (1DL448)
Uppsala University – Autumn 2017
Report for Assignment n / the Project

Clara CLEVER Whiz KIDD
4th October 2017

This document shows the ingredients of a good homework report for the M4CO course. Its \LaTeX{} source code exemplifies almost everything you need to know about \LaTeX{} in order to typeset a professional-looking homework report (for the M4CO course): use it as a starting point for imitation and delete everything irrelevant. The \LaTeX{} source code of this document also contains useful comments. The usage of \LaTeX{} is optional, but highly recommended, for reasons that will soon become clear to those who have never used it before; any learning time is outside the budget of this course, but will hugely pay off, if not in this course then in the next course(s) you take and when writing your BSc, MSc, or PhD thesis.

Address each task of each problem, using the numbering and the ordering in which they appear in the homework statement:

- For each task requiring a model, follow the advice and structure of Section 1.D.
- For each task requiring an evaluation, follow the structure of Section 1.E.

Delete all unnecessary text from this document.

1 The Magic Square Problem

All experiments were run under Linux OpenSuse 11.3 (64 bit) on an Intel Core i7 950 of 3.07 GHz with an 8 MB L2 cache and a 3 GB RAM.\footnote{If different hardware was used for different tasks, then replicate a paragraph like this one within each relevant sub-section.}

1.A Answer for the First Task
(Write your answer for the first task here.)

1.B Answer for the Second Task
...

1.C Use Appropriate Titles for the Subsections
(Remember that this document is a guideline, so that you should change its structure and text when appropriate.)

\footnote{Hint: Under Linux, do \texttt{more /proc/cpuinfo} to find this information. Mac OS X users find this information via “About this Mac” in the Apple menu.}
1.D  Model for the Magic Square Problem

A MiniZinc [1] model must include comments that explain:

- the parameters that are not part of the possibly provided skeleton code;
- the decision variables;
- the problem constraints;
- the objective;
- the redundant decision variables, if any;
- the channelling constraints, if any;
- the implied constraints, if any; and
- the symmetry-breaking constraints, if any.

**The quality of the model comments is considered while grading.** See Listing [1] for an example of sufficiently good model comments; please include your model in your report in addition to submitting it as a separate file.

1.D.1 Description

(Provide, in the report, a more detailed explanation why each channelling, implied, and symmetry-breaking constraint is correct. Use any combination of in-lined code, mathematical notation, and plain English to explain your constraints.)

**Channelling Constraints.** Our model has no channelling constraints, because it has no redundant decision variables.

**Implied Constraints.** Since the sum of each row must be equal to the magic sum, and since all values from 1 to \( n^2 \) must occur exactly once in the magic square, it is implied that the sum of the rows is the sum of the entire magic square: \( n \times \text{magicSum} = \text{sum}(1..n^2) \). Since \( n \) is a parameter, this implied constraint actually fixes the decision variable \( \text{magicSum} \), namely to \( \text{sum}(1..n^2) \mod n \), that is to \( (n^2\times n^2+n) \div 2 \). We reckon this constraint is useful to solvers of all technologies and hence do not flag it using the `implied_constraint` predicate.

**Symmetry-Breaking Constraints.** A magic square has rotation and reflection symmetries:

- The rotation symmetries for 90, 180, and 270 degrees and the reflection symmetries for the horizontal axis, vertical axis, and up-right diagonal can be broken by requiring the top-left corner of the magic square to be smaller than the other corners.

- The reflection symmetry for the down-right diagonal can be broken by requiring the bottom-left corner to be smaller than the top-right corner.

1.D.2 Implementation

The described model, with the comments indicated above, is uploaded as file `magic-square.mzn`. 
Listing 1: A MiniZinc model for the Magic Square problem

```plaintext
include "globals.mzn";

int: n; % the width (and height) of the magic square
set of int: N = 1..n; % the index range for the rows and columns

array[N,N] of var 1..n*n: magic; % magic[r,c] = d iff the magic square has value d at row r and column c
var int: magicSum; % the sum of each row, column, and major diagonal

% All values in the magic square are different:
constraint all_different([magic[r,c] | r, c in N]);

% The sum of each row is equal to the magic sum:
constraint forall(r in N)(sum([magic[r,c] | c in N]) = magicSum);

% The sum of each column is equal to the magic sum:
constraint forall(c in N)(sum([magic[r,c] | r in N]) = magicSum);

% The sum of each major diagonal is equal to the magic sum:
constraint sum([magic[i,i] | i in N]) = magicSum;
constraint sum([magic[n+1-i,i] | i in N]) = magicSum;

% Each row sums up to the magic sum, there are n rows, and their sum is equal to the sum of the entire magic square:
constraint n * magicSum = sum(1..n*n); % implied constraint

% Break the three rotation symmetries and three of the four reflection symmetries by requiring the top−left corner to be smaller than the other corners:
constraint symmetry_breaking_constraint(

  magic[1,1] < magic[1,n] /
  magic[1,1] < magic[n,1] /
  magic[1,1] < magic[n,n]
);

% Break the down−right−diagonal reflection symmetry by requiring the bottom−left corner to be smaller than the top−right corner:
constraint symmetry_breaking_constraint(magic[n,1] < magic[1,n]);

solve satisfy;

output ["Magic sum: ", show(magicSum), "\n"] ++
  [show(magic[r,c]) ++ if c=n then "\n" else " " endif | r,c in N];
```
Compilation and Running Instructions. In order to compile and run `magic-square.mzn` one must supply a value for the parameter `n` using the `-D` option in the command line. Assuming that `mzn-backend` is one of the backends provided for the course, the model can be compiled and run for `n=5` by the command `mzn-backend magic-square.mzn -D "n=5"`

Sample Test-Run Commands.

```
$> ./mzn-oscar-cbls magic-square.mzn -D "n=5"
Magic sum: 65
1 9 10 24 21
12 19 3 23 8
13 11 2 22 17
16 4 14 25 6
15 7 5 20 18
% time from start: 1301
-------
```

1.E Evaluation of the Magic Square Model

We have chosen the backends for Gecode, Chuffed, OscaR, Gurobi, Yices, and MinisatID. (Choose, without justification, any backends you want, as long as you cover all the considered technologies.) Table \ref{tab:results} gives the results for various values of `n` on a mystery minimisation version of our magic-square model; the results are completely made up, in order to illustrate as many particular cases as possible. The time-out was 600 seconds.

We observe that the chosen MIP and SMT backends do not scale for larger instances. The chosen CP and LCG backends prove optimality on the smaller instances, but are very far from finding the best objective value found by the CBLS backend. Recall that the latter can usually not prove the optimality of its solutions and thus usually runs until timing out.

(In other words, refer to and explain the table(s), draw some conclusions from the experiment, and analyse.)
<table>
<thead>
<tr>
<th>Technology</th>
<th>CP</th>
<th>LCG</th>
<th>CBLS</th>
<th>MIP</th>
<th>SMT</th>
<th>Hybrid</th>
</tr>
</thead>
<tbody>
<tr>
<td>Solver</td>
<td>Gecode</td>
<td>Chuffed</td>
<td>OscaR</td>
<td>Gurobi</td>
<td>Yices</td>
<td>MinisatID</td>
</tr>
<tr>
<td>Instance</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>n=9</td>
<td>5</td>
<td>12.3</td>
<td>5</td>
<td>45.6</td>
<td>5</td>
<td>9.1</td>
</tr>
<tr>
<td>n=16</td>
<td>11</td>
<td>89.0</td>
<td>11</td>
<td>123.4</td>
<td>11</td>
<td>56.7</td>
</tr>
<tr>
<td>n=25</td>
<td>22</td>
<td>123.4</td>
<td>24</td>
<td>567.8</td>
<td>17</td>
<td>90.1</td>
</tr>
</tbody>
</table>

Table 1: Results on a mystery minimisation version of our magic-square model. The ‘sol’ column gives the solution status: give ‘unsat’ if the instance is unsatisfiable, otherwise ‘sat’ for a CSP, the best objective value for a COP, or ‘none’ if no feasible solution was found before timing out or crashing. In the ‘time’ column, the time to the first solution, for a CSP, or to the best solution found, for a COP, is given in seconds, but is ‘t/o’ if timing out, or ‘crash’ upon any other error (see the Crash Report below). Times and objective values are bold-faced when a proof of optimality was obtained before timing out; the actual time until the conclusion of the proof of optimality is not reported.
**Hint:** In order to save a lot of time, it is very important that you write a script that conducts the experiments for you and directly generates a result table (see the LATEX source code of Table [1]), which is automatically imported (rather than manually copied) into your report: each time you change the model, it suffices to re-run that script and re-compile your report, in your absence, without any tedious number copying!

1.F Answer for Another Task

...

1.G Alternative Model for the Magic Square Problem
1.G.1 Description
Channelling Constraints. ...

Implied Constraints. ...

Symmetry-Breaking Constraints. ...

1.G.2 Implementation
...

1.H Answer for Yet Another Task
...

Feedback

Please write a paragraph, which will not be graded, describing your experience with this assignment: which aspects were too difficult or too easy, and which aspects were interesting or boring? (Do not forget that optimisation is a highly experimental discipline, so conducting a lot of experiments and reporting on their results is inevitable; also recall the hint above.) This may help us improve the course for the next year.

Intellectual Property

We certify that our report and all its uploaded attachments were produced solely by our team, except where explicitly stated otherwise and clearly referenced, that each teammate can individually explain any part starting from the moment of submitting our report, and that our report and attachments are not and will not be freely accessible on a public repository.

References

Crash Report

For instances where no result is reported, due to a solver crashing or the MiniZinc toolchain failing to compile the model, please include an error message, if there is any. For example:

**largecumulative.mzn** For the larger instances, OscaR/CBLS crashes with the following exception, which seems to be caused by the JVM not being allocated enough heap space.

```plaintext
>./fzn-oscar-cbls -s -t 300 /tmp/tmp.fzn
Exception in thread "main" java.lang.OutOfMemoryError: Java heap space
```
Checklist before Submitting

In order to protect yourself against an unnecessary loss of points, and in order to show both self-respect and respect for the human reader of your report, please use the following checklist before submitting:

- Crosscheck your report against the homework instructions.
- Spellcheck all documents, including the comments in the MiniZinc source code.
- Proofread, if not grammar-check, your report at least once per teammate.

More \LaTeX{} and Technical Writing Advice

Unnumbered itemisation (only to be used when the order of the items does not matter)

- Unnumbered displayed formula:
  \[ E = m \cdot c^2 \]

- Numbered displayed formula (which is cross-referenced somewhere):
  \[ E = m \cdot c^2 \] (1)

- Formula — the same as formula (1) — spanning more than one line:
  \[
  E = m \cdot c^2
  \]

Numbered itemisation (only to be used when the order of the items does matter):

1. First do this.
2. Then do that.
3. If we are not finished, then go back to Step[2] else stop.


Use \verb|\textit{...}| in mathematical mode for each multiple-letter identifier in order to avoid typesetting the identifier like the product of single-letter ones. For example, note the typographic difference between the identifier \verb|WL|, obtained through \verb|$\textit{WL}$|, and the product \verb|WL|, where there is a small space between the \verb|W| and the \verb|L|, obtained through \verb|$WL$|.

[2]Use footnotes very sparingly, and note that footnote pointers are never preceded by a space and always glued immediately behind the punctuation, if there is any.
Figure 1: A binary search tree, a binary min-heap, and a binomial tree of rank 3

Algorithm 1: Silly algorithm

Do not use programming-language-style lower-ASCII notation (such as ! for negation, && for conjunction, || for disjunction, and the equality sign = for assignment) in algorithms or formulas (but rather use ¬ or not, ∧ or & or and, ∨ or or, and ← or :=, respectively), as this testifies to a very strong confusion of concepts.

Figures can be imported with \includegraphics or drawn inside the \LaTeX source code using the highly declarative notation of the tikz package: see Figure 1 for sample drawings. It is perfectly acceptable in this course to include scans or photos of drawings that were carefully done by hand.

Algorithms can be typeset as pseudo-code as exemplified in Algorithm 1: study its \LaTeX source code.

If you are not sure whether you will stick to your current choice of notation or terminology, then introduce a new (possibly parametric) command. For instance, upon

\newcommand{\Cardinality}[1]{\left\lvert#1\right\rvert}

the formula $\Cardinality{S}$ typesets the cardinality of set $S$ as $|S|$ with autosized vertical bars and proper spacing, but upon changing the definition of that parametric command to

\newcommand{\Cardinality}[1]{\# #1}

and recompiling, the formula $\Cardinality{S}$ typesets the cardinality of set $S$ as $\#S$. Similarly, upon

\newcommand{\MiniZinc}{\textit{Mini-Zinc}}

the text \MiniZinc typesets into MiniZinc, hyphenation being only possible in the middle, but upon changing the definition of that non-parametric command to
and recompiling, the text \texttt{MiniZinc} typesets into MiniZinc. You can thus obtain an arbitrary number of changes in the document with a constant-time change in its source code, rather than having to perform a linear-time find-and-replace operation within the source code, which is painstaking and error-prone. The imported file macros.tex has a lot of useful predefined commands about mathematics, CP, MiniZinc, modelling, and algorithms.

Use commands on positioning (such as \hspace, \vspace, and \noindent) and appearance (such as \small for reducing the font size, and \textit for italics) very sparingly, and ideally only in (parametric) commands, as the very idea of mark-up languages such as \LaTeX{} is to let the class designer (usually a trained professional typesetter) decide on where things appear and how they look. For instance, \textbf (for emphasis) compiles (outside italicised environments, such as \emph{theorem}) into \textit under the \texttt{article} class used for this document, but it may compile into \textbf\emph under some other class.

If you do not (need to) worry about \textit{how} things look, then you can fully focus on \textit{what} you are trying to express!

Note that \textit{no} absolute numbers are used in the \LaTeX{} source code for any of the references inside this document. For ease of maintenance, \texttt{\label} is used for giving a label to something that is automatically numbered (such as an algorithm, equation, figure, footnote, item, line, section, subsection, or table), and \texttt{\ref} is used for referring to a label. An item in the bibliography file is referred to by \texttt{\cite} instead. Upon changing the text, it suffices to recompile, once or twice, and possibly to run BibTeX again, in order to update all references consistently.

Prefer Section∼\texttt{\ref{sec:magicS}} over Section \texttt{\ref{sec:magicS}}, using the non-breaking space (typeset as ∼) instead of the space, as this gives “Section 1” instead of “Section 1” and thereby avoids that a cross-reference is spread across a line break, as happened in the previous line: this is considered poor typesetting.

The rules of English for how many spaces to use before and after various symbols are given in Table 3. Beware that they may be very different from the rules in your native language.

Feel free to report to the head teacher any other features that you would have liked to see discussed and exemplified in this template document.
<table>
<thead>
<tr>
<th>Topic</th>
<th>\LaTeX{} code</th>
<th>Appearance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Greek letter</td>
<td>$\Theta, \Omega, \epsilon$</td>
<td>$\Theta, \Omega, \epsilon$</td>
</tr>
<tr>
<td>multiplication</td>
<td>$m \cdot n$</td>
<td>$m \cdot n$</td>
</tr>
<tr>
<td>division</td>
<td>$\frac{m}{n}, m \div n$</td>
<td>$\frac{m}{n}, m \div n$</td>
</tr>
<tr>
<td>rounding down</td>
<td>$\left\lfloor n \right\rfloor$</td>
<td>$\left\lfloor n \right\rfloor$</td>
</tr>
<tr>
<td>rounding up</td>
<td>$\left\lceil n \right\rceil$</td>
<td>$\left\lceil n \right\rceil$</td>
</tr>
<tr>
<td>binary modulus</td>
<td>$m \mod n$</td>
<td>$m \mod n$</td>
</tr>
<tr>
<td>unary modulus</td>
<td>$m = n \mod \ell$</td>
<td>$m = n \mod \ell$</td>
</tr>
<tr>
<td>root</td>
<td>$\sqrt{n}, \sqrt[3]{n}$</td>
<td>$\sqrt{n}, \sqrt[3]{n}$</td>
</tr>
<tr>
<td>exponentiation, superscript</td>
<td>$n^{i}$</td>
<td>$n^{i}$</td>
</tr>
<tr>
<td>subscript</td>
<td>$n_{i}$</td>
<td>$n_{i}$</td>
</tr>
<tr>
<td>overline</td>
<td>$\overline{n}$</td>
<td>$\overline{n}$</td>
</tr>
<tr>
<td>base 2 logarithm</td>
<td>$\log_2 n$</td>
<td>$\log_2 n$</td>
</tr>
<tr>
<td>base b logarithm</td>
<td>$\log_b n$</td>
<td>$\log_b n$</td>
</tr>
<tr>
<td>binomial</td>
<td>$\binom{n}{k}$</td>
<td>$\binom{n}{k}$</td>
</tr>
<tr>
<td>sum</td>
<td>$\sum_{i=1}^{n} i$</td>
<td>$\sum_{i=1}^{n} i$</td>
</tr>
<tr>
<td>numeric comparison</td>
<td>$\leq, &lt;, =, \neq, &gt;, \geq$</td>
<td>$\leq, &lt;, =, \neq, &gt;, \geq$</td>
</tr>
<tr>
<td>non-numeric comparison</td>
<td>$\prec, \nprec, \preceq, \succeq, \succ$</td>
<td>$\prec, \nprec, \preceq, \succeq, \succ$</td>
</tr>
<tr>
<td>extremum</td>
<td>$\min, \max, +\infty, \bot, \top$</td>
<td>$\min, \max, +\infty, \bot, \top$</td>
</tr>
<tr>
<td>function</td>
<td>$f: A \to B, \circ, \mapsto$</td>
<td>$f: A \to B, \circ, \mapsto$</td>
</tr>
<tr>
<td>sequence, tuple</td>
<td>$\langle a, b, c \rangle$</td>
<td>$\langle a, b, c \rangle$</td>
</tr>
<tr>
<td>set</td>
<td>${a, b, c}, \emptyset, \mathbb{N}$</td>
<td>${a, b, c}, \emptyset, \mathbb{N}$</td>
</tr>
<tr>
<td>set membership</td>
<td>$\in, \not\in$</td>
<td>$\in, \not\in$</td>
</tr>
<tr>
<td>set comprehension</td>
<td>${i \mid 1 \leq i \leq n}$</td>
<td>${i \mid 1 \leq i \leq n}$</td>
</tr>
<tr>
<td>set operation</td>
<td>$U, \cap, \setminus, \times$</td>
<td>$U, \cap, \setminus, \times$</td>
</tr>
<tr>
<td>set comparison</td>
<td>$\subset, \subseteq, \not\supset$</td>
<td>$\subset, \subseteq, \not\supset$</td>
</tr>
<tr>
<td>logic quantifier</td>
<td>$\forall, \exists, \nexists$</td>
<td>$\forall, \exists, \nexists$</td>
</tr>
<tr>
<td>logic connective</td>
<td>$\land, \lor, \neg, \Rightarrow$</td>
<td>$\land, \lor, \neg, \Rightarrow$</td>
</tr>
<tr>
<td>logic</td>
<td>$\models, \equiv, \vdash$</td>
<td>$\models, \equiv, \vdash$</td>
</tr>
<tr>
<td>miscellaneous</td>
<td>$&amp;,#, \approx, \sim, \ell$</td>
<td>$&amp;,#, \approx, \sim, \ell$</td>
</tr>
<tr>
<td>dots</td>
<td>$\ldots, \cdots, \vdots, \ddots$</td>
<td>$\ldots, \cdots, \vdots, \ddots$</td>
</tr>
<tr>
<td>dots (context-sensitive)</td>
<td>$1, \ldots, n; 1 + \ldots + n$</td>
<td>$1, \ldots, n; 1 + \ldots + n$</td>
</tr>
<tr>
<td>parentheses (autosizing)</td>
<td>$\left(m^{n^k}\right), (m^{n^k})$</td>
<td>$\left(m^{n^k}\right), (m^{n^k})$</td>
</tr>
<tr>
<td>identifier of &gt; 1 character</td>
<td>$\text{\texttt{mathit{identifier}}}$</td>
<td>$\text{\texttt{identifier}}$</td>
</tr>
<tr>
<td>hyphen, n-dash, m-dash, minus</td>
<td>$-,-,-,-,$</td>
<td>$-,-,-,-,$</td>
</tr>
</tbody>
</table>

Table 2: The typesetting of elementary mathematics. Note very carefully when italics are used by \LaTeX{} and when not, as well as all the horizontal and vertical spacing performed by \LaTeX{}.

<table>
<thead>
<tr>
<th>number of spaces after</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>number of spaces before</td>
<td>0</td>
<td>/</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>(</td>
</tr>
</tbody>
</table>

Table 3: Spacing rules of English