

# 50.021 – AI

Kwan Hui, Dorien and Alex

## Week 3 Coding - Constraint Satisfaction Problems

[The following notes are compiled from various sources such as textbooks, lecture materials, Web resources and are shared for academic purposes only, intended for use by students registered for a specific course. In the interest of brevity, every source is not cited. The compiler of these notes gratefully acknowledges all such sources. ]

**Due: 16th June, 11:59pm**

Submission: via eDimension

### 1 Coding: Semi-Magic Square

Consider a  $3 \times 3$  array, each of whose entries can be either 1, 2 or 3. We want to find an assignment to each of the entries so that the entries in each row, in each column and in one of the diagonals are different. Note that this will also ensure that these row, column and diagonals add up to 6 ( $1 + 2 + 3$ ). But, note that the "adding to 6" constraint is not a "binary constraint", that is, it involves more than three variables. However, the constraint that each pair of values in the row, column or diagonal be different is a "binary constraint".

You can find a description of the Semi-Magic Square problem in the attached slides (constraint05.pdf), in particular in Slide 14 (reproduced here).

## A New CSP (where fancier propagation is possible)

- The semi magic square
- Each variable can have value 1, 2 or 3

$V_1$	$V_2$	$V_3$	← This row must sum to 6
$V_4$	$V_5$	$V_6$	← This row must sum to 6
$V_7$	$V_8$	$V_9$	← This row must sum to 6
↑ This column must sum to 6	↑ This column must sum to 6	↑ This column must sum to 6	↖ This diagonal must sum to 6

Slide 14

Your tasks are as follows:

1. Implement a CSP that captures this problem. Use the file `semi-magic.py` in the code distribution. It also imports `csp.py`. You should use the variable names in the image above in your CSP formulation.
2. Experiment solving this problem with the various solution methods, ranging from pure backtracking and its various enhancements/heuristics such as variable and value orderings, forward checking, etc.
3. Look at the number of assignments attempted by each algorithm, that should give you some idea of the effectiveness of the methods on this problem. Elaborate on your findings from these results, i.e., what you understand.