Basic Numberjack Tutorial

Adapted from Hebrard et al.'s AAAI 2010 tutorial and parts of the Numberjack website

CS 175

April 5, 2011
1. Introduction

2. Intro to Python

3. Modeling in Numberjack

4. Examples
   - N-Queens Problem
   - Magic Squares
   - Combinatorial Auctions

5. Conclusion
1 Introduction

2 Intro to Python

3 Modeling in Numberjack

4 Examples
   - N-Queens Problem
   - Magic Squares
   - Combinatorial Auctions

5 Conclusion
What is Numberjack?

- A platform for constraints
- Written in Python - a front-end to C++-based solvers
- Excellent for rapidly trying out models
What is Numberjack?

- A platform for constraints
- Written in Python - a front-end to C++-based solvers
- Excellent for rapidly trying out models
- "Cuts your exponential search tree into logs"
1. Introduction

2. Intro to Python

3. Modeling in Numberjack

4. Examples
   - N-Queens Problem
   - Magic Squares
   - Combinatorial Auctions

5. Conclusion
Overview of Python

- Scripting language
- Supports classes, objects, etc.
- Duck-typing
Variables and Functions

Variables

\[ a = 2 \]

- No need to declare the variable
- Variables are untyped

Functions

```python
def double(a):
    return a * 2
```

- Functions are also not typed
- Indentations based on whitespace and are part of the syntax
Lists and Tuples

Lists

```python
foo = [1, 4, 5, 10, 2]
bar = ["this", "is", "a", "list"]
```

Tuples

```python
triplet = (1, 2, 3)
course = ("CS", 175)
```

- Again, types don’t matter even within lists and tuples
Control

```python
if <boolean_exp>:
    do_stuff()

while <boolean_exp>:
    do_stuff()
```
For Loops

For loops in C/C++/Java

```plaintext
for (int i = 0; i < n; ++i) {
    do_stuff(i)
}
```

For loops in Python

```plaintext
for i in range(n):
    do_stuff(i)
```
More Fun with For Loops

```python
for element in list:
    do_stuff_with(element)

teamProjects = [(0, "Asteroid Simulation"),
                 (1, "Scrabble"),
                 (2, "Poker")]

for teamNumber, project in teamProjects:
    print "Team", teamNumber, ":", project

Team 0 : Asteroid Simulation
Team 1 : Scrabble
Team 2 : Poker
```
List Comprehensions

A very useful feature!

```python
>>> range(4)
[0, 1, 2, 3]
>>> [x * 2 for x in range(4)]
[0, 2, 4, 6]
>>> [x * 2 for x in range(4) if x >= 2]
[4, 6]
```

Generally,

```
[<expression> for x in <Iterable> (if <condition>)]
```
1 Introduction

2 Intro to Python

3 Modeling in Numberjack

4 Examples
   - N-Queens Problem
   - Magic Squares
   - Combinatorial Auctions

5 Conclusion
Overview

- Constructs
  - Variables
  - Constraints
  - Model

- A common API to interface with back-end solvers
### Variables

```python
# binary variable
Variable()

# domain from 0 to N-1
Variable(N)

# domain from L to U
Variable(L, U)

# domain specified by a list
Variable(list)
```

Useful method (used after a solution has been found)

```python
get_value()
```
Variables

More constructors:

# create a list of N binary variables
VarArray(N)

# create a list of N variables with domains from 0 to D−1
VarArray(N, D)

# create a list of N variables with domains from L to U
VarArray(N, L, U)
Variables

...and even more constructors:

```python
# create a matrix of M x N binary variables
m = Matrix(M, N)
# create a matrix of M x N variables with domains from L to U
m = Matrix(M, N, L, U)
```

Special operators

```python
# Return a VarArray containing all of the elements of the Matrix
m.flat
# Return a list of VarArrays corresponding to each row
m.row
# Return a list of VarArrays corresponding to each column
m.col
```
Constraints

- **Arithmetic operators on variables**
  
  
  \[
  \begin{align*}
  &x > y \\
  &x == y + 2 \\
  &m[1][4] != n[4][3]
  \end{align*}
  \]

- **Global constructors**

  \[
  \begin{align*}
  &\text{AllDiff([a, b, c, d, e])} \\
  &\text{AllDiff(myVarArray)} \\
  &\text{AllDiff(myMatrix)} \\
  &\text{Sum([a, b, c, d]) >= e}
  \end{align*}
  \]
Model

- Used to collect the constraints together to define a problem

- Constructors

  # empty model
  model = Model()

  # model with constraints
  model = Model(constraints,...)

- Adding more constraints

  model.add(constraints)
  #or
  model += constraints
Using a Solver

- Different solvers available (Mistral, MiniSat, Walksat)

- Methods

  # Get a solver to solve the given problem specified
  # by the model,
  solver = model.load('nameOfSolver')
  # attempts to solve the problem
  solver.solve()
  # for search–based solvers only (to generate multiple solutions)
  solver.startNewSearch()
  while solver.getNewSolution():
    # do something with solution

- Results are stored in the Variable objects
Outline of Usage

- Specify variables
- Specify constraints over those variables
- Construct a model with the constraints
- Construct the solver using that model
- Call solve() and extract results from Variables using get_value()
Outline of Usage

- Specify variables
- Specify constraints over those variables
- Construct a model with the constraints
- Construct the solver using that model
- Call `solve()` and extract results from Variables using `get_value()`
- Can alternatively use the print statement on Variables directly to output their values
N-Queens Problem

1. Introduction

2. Intro to Python

3. Modeling in Numberjack

4. Examples
   - N-Queens Problem
   - Magic Squares
   - Combinatorial Auctions

5. Conclusion
Problem Definition

- Place queens on the chessboard such that no two queens are attacking each other
N-Queens Problem

Modeling

- What are the variables/domains of variables for the 4-queens problem presented?
What are the variables/domains of variables for the 4-queens problem presented?

How about in general for the N-queens problem?
N-Queens Problem

Modeling

- What are the variables/domains of variables for the 4-queens problem presented?
- How about in general for the N-queens problem?
- What constraints do we need?
1. Introduction

2. Intro to Python

3. Modeling in Numberjack

4. Examples
   - N-Queens Problem
   - Magic Squares
   - Combinatorial Auctions

5. Conclusion
Problem Definition

Given an \( N \times N \) square, place numbers ranging from 1 to \( N^2 \) such that each row, column, and diagonal has the same sum.
Magic Squares

Modeling

- Same questions as before...(variables, domains, constraints?)
Magic Squares

Modeling

- Same questions as before... (variables, domains, constraints?)
- Which Numberjack Variable constructor seems appropriate for this?
1 Introduction

2 Intro to Python

3 Modeling in Numberjack

4 Examples

- N-Queens Problem
- Magic Squares
- Combinatorial Auctions

5 Conclusion
Problem Definition

<table>
<thead>
<tr>
<th>Items</th>
<th>Bid Amount</th>
<th>Variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>A, B</td>
<td>10</td>
<td>$x_0$</td>
</tr>
<tr>
<td>A, C</td>
<td>20</td>
<td>$x_1$</td>
</tr>
<tr>
<td>B, D</td>
<td>20</td>
<td>$x_2$</td>
</tr>
<tr>
<td>B, C, D</td>
<td>25</td>
<td>$x_3$</td>
</tr>
<tr>
<td>A</td>
<td>14</td>
<td>$x_4$</td>
</tr>
</tbody>
</table>

Choose bids such that sets of items across bids are disjoint
## Problem Definition

<table>
<thead>
<tr>
<th>Items</th>
<th>Bid Amount</th>
<th>Variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>A, B</td>
<td>10</td>
<td>$x_0$</td>
</tr>
<tr>
<td>A, C</td>
<td>20</td>
<td>$x_1$</td>
</tr>
<tr>
<td>B, D</td>
<td>20</td>
<td>$x_2$</td>
</tr>
<tr>
<td>B, C, D</td>
<td>25</td>
<td>$x_3$</td>
</tr>
<tr>
<td>A</td>
<td>14</td>
<td>$x_4$</td>
</tr>
</tbody>
</table>

- Choose bids such that sets of items across bids are disjoint
- ...such that the selection maximizes the revenue
Problem Definition

<table>
<thead>
<tr>
<th>Items</th>
<th>Bid Amount</th>
<th>Variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>A, B</td>
<td>10</td>
<td>$x_0$</td>
</tr>
<tr>
<td>A, C</td>
<td>20</td>
<td>$x_1$</td>
</tr>
<tr>
<td>B, D</td>
<td>20</td>
<td>$x_2$</td>
</tr>
<tr>
<td>B, C, D</td>
<td>25</td>
<td>$x_3$</td>
</tr>
<tr>
<td>A</td>
<td>14</td>
<td>$x_4$</td>
</tr>
</tbody>
</table>

- Choose bids such that sets of items across bids are disjoint
- ...such that the selection maximizes the revenue
- Different from constraint satisfaction...known as constraint optimization
- In addition to constraints, we need to specify an objective function
Combinatorial Auctions

Modeling

- The number of variables is given this time, but what are the domains?
- What is the objective function?
- What are the constraints?
Conclusion

- Rapid prototyping of problems
- Easy to test out different solvers
- Numberjack website: http://numberjack.ucc.ie (also linked from the course page)