

离散优化建模：习题课0

第一步

1 简介

本次的习题课是关于如何使用MiniZinc以及尝试一些简单的模型。

1.1 Hello World - hello.mzn

建立一个MiniZinc模型**hello.mzn** 用来输出“Hello World”!

用下面的命令进行测试：

```
minizinc hello.mzn
```

1.2 输入与输出 - io.mzn

建立一个MiniZinc模型**io.mzn**，它可以通过一个数据文件输入整型参数 n ，然后输出它的值。

用以下命令进行测试：

```
minizinc io.mzn -D"n = d;"
```

d 输入不同的值，模型将输出不同的值。

1.3 简单的决策 - x110.mzn

建立一个MiniZinc模型文件**x110.mzn**。这个模型有一个取值范围为1到10的决策变量 x 。它会输出该变量的具体数值。

用以下命令进行测试：

```
minizinc x110.mzn -a
```

模型应该把 x 的所有解都输出出来。

1.4 简单的优化 - xopt.mzn

建立一个MiniZinc的模型文件**xopt.mzn**。这个模型有一个取值范围为1到10的决策变量 x ，而且有一个约束保证 x 可以被4整除。它会输出令 $(x - 7)^2$ 的值最小化时 x 的值。

用以下命令进行测试：

```
minizinc xopt.mzn
```

模型应该把最优解输出出来。也可以用以下命令进行测试：

```
minizinc xopt.mzn -a
```

这时模型应该把所有它在求解过程中的解输出出来。注意`-a`选项在满足问题（输出问题所有的可行解），和在优化问题（输出求解过程中的所有解）中的不同应用。

问题 假设你不能使用`mod`函数，你还可以如何建模令到 x 可以被4整除？

1.5 数组 - array.mzn

定义一个MiniZinc模型文件`array.mzn`。这个模型需要输入一个整型参数 n ， n 用于决定数组变量 x 的长度。 x 里面的每一个值的取值范围为0到9。约束这个数组令到它的各个数值的总和等于各个数值的乘积。最后模型把该数组输出出来。

用以下命令进行测试：

```
minizinc array.mzn -a
```

模型应该输出 x 的所有可行解。

新增一个约束令到数组中的数字是非递增的，也就是 $x[1] \leq x[2] \leq \dots \leq x[n]$ 。这个约束可以减少相似解的个数。这一个对称性破缺的例子。这在建模中非常有用。

你的模型可以求解的数组有多大？你认为这种情况为什么会发生？

1.6 序列 - seq.mzn

定义一个MiniZinc模型文件`seq.mzn`。这个模型需要输入一个整型参数 n ， n 用于定义数组变量 x 的长度。 x 里面的每一个值得取值范围为0到3。约束这个数组中的数字令到第一个数字是0，最后一个数字是3。同时约束这个数组 x ，令到在 x 中位置可以被3整除的数字大于等于2。模型需最大化数组 x 中各个数字的总和。模型的输出格式为：总和 = 数组中的各个数值。例如，对于一个解，其中 x 长度为5的数组`[0,1,2,0,3]`，输出为：

```
6 = [0,1,2,0,3]
```

对于 n 的不同数值 d ，用以下命令进行测试：

```
minizinc seq.mzn -D"n = d;"
```

至少测试 d 的数值为3到9的情况。