

# 离散优化建模：作业二

## 战斗准备

### 1 问题描述

刘备要准备一场对黄巾军的战斗，为此他必须选择一批勇士参与这场战斗。战斗部队需要有  $l$  到  $u$  名勇士。每一个勇士都是某个宗族的成员，而且有的宗族之间的人相互敌视。在战斗部队中，不能有多于  $m$  对勇士的宗族是互相敌视的。战斗部队的战斗力是由部队中全体勇士的战斗力之和决定的，而问题的目标是要使战斗部队的战斗力最大化。

### 2 数据格式说明

战斗准备问题的输入是名为 `data/raid_p.dzn` 数据文件，其中  $p$  是问题的序号。文件中， $l$  是战斗部队的最小人数， $u$  是战斗部队的最大人数，部队中可容许的最多  $m$  对勇士互相敌视。`WARRIOR` 是一个枚举类型，它代表了不同勇士，其中包括一个 *dummy* 的勇士，它的战斗力比其他勇士都要低。*strength* 是每个勇士的战斗力 ( $strength[dummy] = 0$ )。`CLAN` 是一个枚举型，它记录了不同的宗族，而 *clan* 定义了不同勇士所来自的宗族 (*dummy* 勇士的宗族没有与其他任何的宗族相互敌视)。*hates* 是一个二维数组，其中的变量取值 0 或 1。它定义了哪些宗族之间相互敌视（这个数组对应的矩阵是对称的）。

例如：

```
l = 4;
u = 6;
m = 2;
WARRIOR = { D, W1, W2, W3, W4, W5, W6, W7, W8, W9, W10 };
dummy = D;
CLAN = { DC, C1, C2, C3, C4 };
strength = [ 0, 9, 6, 4, 10, 8, 3, 4, 2, 7, 1 ];
clan = [ DC, C1, C1, C1, C2, C2, C2, C3, C3, C4, C4 ];
hates = [| 0, 0, 0, 0, 0, 0
          | 0, 0, 1, 0, 0, 0
          | 0, 1, 0, 0, 0, 1
          | 0, 0, 0, 0, 0, 0
          | 0, 0, 1, 0, 0, 0 |];
```

你的模型输出应该给出战斗部队的组成和对应的目标数值。比如，模型的输出可以是：

```
raid = { W1, W4, W5, W7, W8 };
obj = 33;
```

注意以上输出中有两对互相敌视的勇士，分别是：W1 跟 W4，W1 跟 W5。模板文件 `raid.mzn` 是用来展示如何读取输入数据的。

你可以使用任何集合的表示形式来代表 `raid` 变量，但需要相应地修改输出语句。你可以用以下输出命令：

```
array[1..n] of var OBJ: s;
output ["s = {" ] ++ [show(s[i]) ++ if i == n then "};\n" else ", " endif | i in 1..n ]
```

来输出一个利用数组表示的固定势集合。你也可以用以下输出命令：

```
array[1..n] of var OBJx: s;
output ["s = {" ] ++ [if fix(s[i]) != dummy then show(s[i])
    ++ if i < n /\ fix(s[i+1]) != dummy then ", " else "" endif
    else "" endif | i in 1..n ] ++ ["};\n"];
```

来输出一个利用数组表示的有限势集合。

### 3 指引

你可以编辑 `raid.mzn` 模型文件来解决上述优化问题。你实现的模型 `raid.mzn` 可以用提供的文件进行测试。在 MINIZINC IDE 中，你可以通过点击 *Run* 按钮在本地测试和运行。或者在命令行中输入

```
mzn-gecode ./raid.mzn ./data/<inputFileName>
```

进行本地测试和运行。两种情况下，你的模型都是用 MINIZINC 进行编译同时用 GECODE 求解器求解。

**参考资料** 你可以在 `data` 文件夹下找到讲义中的几个问题实例（的数据文件）。

**提交作业** 这次的作业包含有 3 个答案提交部分和 1 个模型提交部分。对于答案提交部分，我们将会提交求解器求解你的模型所得到的最好 / 最后的答案，然后检查它的正确性和得分。对于模型提交部分，我们将会提交你的模型文件 (`.mzn`) 然后用一些隐藏的数据文件来做进一步检查。

在 MINIZINC IDE，点击 *coursera* 图标可以用于提交作业。若采用命令行方式，`submit.py` 可以用于提交作业。无论采用那种方法，你都需要根据本指引中的要求完成作业各部分的 MiniZinc

模型。你可以多次提交，最终作业分数是你的最高的一次。<sup>1</sup>作业的打分过程可能需要几分钟，请耐心等待。你可以在课程网站上的编程作业 版块查看你的作业提交状况。

## 4 软件要求

为了完成作业,你需要安装MINIZINC 2.1.x和GECODE 5.0.x 求解器。这些软件都会包含在MINIZINC IDE 2.1.2 (<http://www.minizinc.org>)的集成版本中。如果你需要通过命令行提交作业，你需要安装Python 3.5.x。

---

<sup>1</sup>答案提交部分并没有次数限制。但是，模型提交部分只能提交有限次。