

Comando

Javier Larrosa

.

1 Introduction

Los altos mandos de un batallón de infantería tienen que escoger a un conjunto de soldados para realizar una serie de misiones y nos piden ayuda para hacerlo de manera óptima.

En una primera reunión nos describen el problema diciendo que cada misión y cada soldado tiene unas características y que el objetivo es maximizar el éxito de las misiones, pero minimizando lo que se pone en riesgo.

2 Especificación

Como la explicación anterior es muy imprecisa, hacemos una segunda reunión para mejorar la especificación del problema. Tras un proceso de preguntas, respuestas, simplificaciones, etc llegamos a la siguiente especificación del problema.

Tenemos n soldados disponibles. El número de soldados que tenemos que asignar a cada misión está acotado entre dos números. Para cada misión tenemos un número que nos determina su dificultad. Como hay soldados más valiosos que otros (más valientes, más hábiles, etc) para cada soldado sabemos su valor codificado con un número entre 0 y 10. Para garantizar el éxito de las misiones, la suma del valor de los soldados que se asignan a cada misión tiene que ser como mínimo igual a su dificultad. En todas las misiones tiene que haber al menos dos soldados que hablen el idioma del enemigo y al menos un experto en telecomunicaciones. Ningún soldado puede estar asignado a más de una misión y tampoco puede haber hermanos asignados en la misma misión.

Tenemos que decidir qué soldados se asignan a cada misión teniendo como objetivo minimizar la suma de valor de los soldados asignados.

3 Modelado con MiniZinc

3.1 Datos

Decidimos que los datos (parametros) de las instancias tienen que ser compatibles con la siguiente declaración:

```

include "globals.mzn";
int: n; % numero de soldados
set of int: SOLD=1..n;
array[SOLD] of 0..10: valor;
set of SOLD: idioma;
set of SOLD: teleco;
int: h; %numero de pares de hermanos
array[1..h,1..2] of SOLD: hermanos;

int: k;% numero de misiones
set of int: MIS= 1..k;
array[MIS] of int: minimo;
array[MIS] of int: maximo;
array[MIS] of int: dificultad;

```