Resolución de problemas de selección de incidencias por coste y beneficio utilizando D-Wave

Ismael Pérez Nieves

GSyA
ismael.perez3@alu.uclm.es

Abstract

La selección de incidentes es un problema complejo que ayuda en gran medida a la organización de una empresa. Si conocemos el tiempo que requerirá resolver estos incidentes y le damos un valor que nos aportaría resolverlo, podemos conocer qué conjunto de incidentes sería más óptimo resolver dado un tiempo máximo. Este problema es NP-hard, pero con la ayuda de la computación cuántica podremos resolver estos problemas con alta eficiencia.

Contents

1 Introduction

La computación cuántica nos permite resolver problemas que antes nos eran imposibles. Combinando la computación clásica con la computación cuántica, podemos crear artefactos software capaces de resolver problemas como la selección de incidentes. Este es el objetivo de D-Wave, que ofrece una libería para el lenguaje de programación Python como middleware para poder acceder a sus computadores cuánticos. Este proceso de resolución de problema se centrará en el proceso de creación de una matriz QUBO (Quadratic Unconstrained Binary Optimization), equivalente al modelo Ising.

2 Quadratic Unconstrained Binary Optimization

El objetivo de la creación de la matriz QUBO es contener la energía asociada a la relación cuadrática entre las variables binarias que representarán nuestros objetos a seleccionar, en este caso incidentes. Una vez creada esta matriz, podremos utilizar un computador cuántico para generar un número definido de posibles soluciones que le darán un valor de 0 o 1 a cada variable binaria. A partir de estos valores, las relaciones cuadráticas seleccionadas sumarán sus energías a la solución. Por último, se seleccionaría la solución con la mayor o menor energía de todas, dependiendo del criterio de selección.

En nuestro caso, minimizaremos la energía de las soluciones. Tendremos que tener esto en cuenta para construir la función que definirá a la función QUBO.

2.1 Función QUBO

La función QUBO es una función algebraica que determinará los valores de la energía de una solución dada. Podemos dividirla en dos partes. Primero tenemos la función de optimización, que será la encargada de seleccionar dentro de las soluciones correctas cuál es la más optima. Para seleccionar cuáles son las soluciones correctas, tenemos la segunda parte de la función que es la parte de penalización o condición.

2.1.1 Función de optimización

En esta función deberemos determinar cuál va a ser nuestra condición de optimización. En otras palabras, cuál va a ser el criterio para decidir qué solución es la más óptima. En el caso del problema de selección de incidencias, sería la solución que más valor nos aportase. Por lo tanto, podemos definirlo como el sumatorio de los valores asociados a cada incidencia seleccionada. Expresado de forma algebraica:

$$\sum_{i=1}^{N}(C_i \cdot x_i)$$

Siendo C el conjunto de valores asociados a las incidencias y N el número de incidencias. Tenemos que tener en cuenta que nuestro objetivo es minimizar la energía y maximizar el valor, por lo que deberemos restar este término en la función final.

2.1.2 Función de penalización

A la hora de construir la función de penalización, tenemos que estar pensando en cuál va a ser la condición que determine que una solución es inválida. En nuestro caso, una solución es inválida cuando el peso total de las incidencias que hemos seleccionado supera el peso máximo permitido. Expresado de forma algebraica:

$$\sum_{i=1}^{N}(W_i \cdot x_i) \geq W_M$$

Sin embargo, no podemos utilizar esta expresión ya que es una inecuación. De alguna forma, tenemos que transformar esta inecuación en una igualdad. En el libro (añadir referencia) se habla de unos procesos llamados Transformation 1 y Transformation 2, que son las que usaremos.

Estas transformaciones consisten en añadir variables de slack a nuestra inecuación de forma que podamos considerarla una igualdad. Estas variables de slack son variables imaginarias que su único objetivo es mantener la igualdad, no tienen ningún significado real. Lo que queremos hacer con las variables de slack es hacer que todos los números iguales o mayores al peso máximo pasen a ser iguales al peso máximo. Esto lo podemos conseguir con:

$$\sum_{i=0}^{N}(W_i \cdot x_i) - S = W_T$$

Siendo S una variable indeterminada positiva que es igual a la diferencia entre el peso máximo y el primer término. Nótese que si el primer término es menor que W, es imposible que esta igualdad se cumpla. Esta variable S la conseguiremos con nuestras variables de slack. Podríamos definir S como:

$$S = \sum_{i=0}^{K}(S_i \cdot y_i)$$

Siendo K la diferencia máxima entre el peso máximo y el primer término de la función de penalización. Sin embargo, esto es ineficiente y redundante. Vamos a utilizar el método de (insertar referencia aquí). Con ello, podemos definir S como:

$$S = \sum_{i=0}^{M-1}(2^i \cdot y_i) + (N + 1 - 2^M)y_M$$

Con todo esto, tenemos nuestra función de optimización:

$$\sum_{i=0}^{N}(W_i \cdot x_i) - \sum_{i=0}^{M-1}(2^i \cdot y_i) + (N + 1 - 2^M)y_M = W_T$$

$$(\sum_{i=0}^{N}(W_i \cdot x_i) - \sum_{i=0}^{M-1}(2^i \cdot y_i) + (N + 1 - 2^M)y_M - W_T)^2$$

Ya solo nos quedaría hacer la identidad notable y programar el QUBO.