ugr | Universidad de Granada

TRABAJO FIN DE GRADO

INGENIERÍA INFORMÁTICA

Aplicación de Metaheurísticas Cuánticas para la Optimización de Cadenas de Servicios en un Modelo de Red 5G

Manual de Instalación

Autor:
Mario Guisado García

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍAS INFORMÁTICA Y DE TELECOMUNICACIÓN

—

Índice

1 Configuración **1**

2 Ejemplo de Ejecución **2**

1. Configuración

Aunque en la memoria de este Trabajo de Fin de Grado ya existe una sección dedicada a la configuración e instalación de bibliotecas necesarias para la ejecución del código desarrollado, en este documento se detallará dicho proceso de nuevo junto con un ejemplo de ejecución.

Los pasos descritos para la configuración del entorno son los mismos facilitados por la propia facultad con motivo de los talleres de computación cuántica llevados a cabo por Fujitsu. En estos talleres, se ejecutaba el código desarrollado desde Google Colab, plataforma que permite usar y compartir cuadernos de Jupyter. Para el desarrollo de este trabajo y debido a la complejidad y magnitud del mismo, era necesario trabajar en un editor de código propio.

Los pasos seguidos para configurar el ordenador en el que se ha trabajado (con sistema operativo Windows 10) son:

- Descargar el instalador de https://www.anaconda.com/downloaddownloads.
- Instalar Anaconda. Anaconda es una distribución gratuita y de código abierto de Python que incluye una gran cantidad de paquetes y herramientas, lo que facilita la gestión de entornos y dependencias.
- Crear un nuevo ”environment” con la version de python apropiada, para ello abrir el Prompt de Anaconda en Windows, o un terminal normal en Linux o Mac y ejecutar la siguiente instrucción: conda create -n “expertcourse”python=3.10.13. Un environment en Anaconda es un entorno aislado que tiene su propia instalación de Python y paquetes específicos lo que facilita el trabajo en diferentes proyectos con diferentes dependencias sin tener conflictos.

- Descargar el fichero dadk-light-3.10.tar.bz2 y colocarlo en la misma ruta que el fichero de requerimientos. Esta biblioteca será la correspondiente al simulador cuántico que se usará en las primeras fases del desarrollo, y no está incluída en Anaconda, por lo que debe descargarse por separado.
- Instalar el fichero de requerimientos con pip install -r "requirements.txt". Este fichero contiene una lista de dependencias necesarias para que el proyecto funcione correctamente. Esta acción instalará todas las dependencias en un único comando.

Llegado a este punto se puede utilizar el editor de código de preferencia (recomendado VS Code) asegurando cambiar el environment en cuestión. El resto de bibliotecas que pudiesen ser necesarias se instalarán usando pip con la forma: pip install *biblioteca*.

2. Ejemplo de Ejecución

Abierto el editor, en este caso VS Code, podremos seleccionar el environment arriba creado:

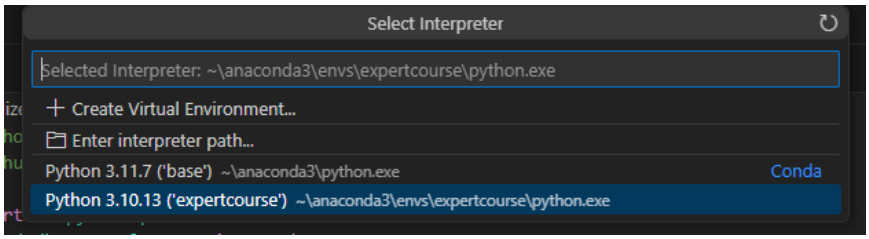

Figura 1: Selección del entorno creado.

El entorno seleccionado será visible en la información que aparece abajo a la derecha:

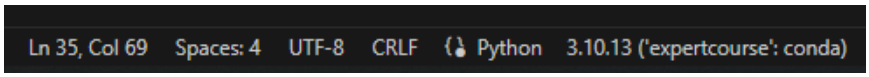

Figura 2: Información disponible en la parte inferior del editor.

Tras esto, ejecutamos el archivo principal *main.py* y a continuación se nos realizarán una serie de preguntas sobre el problema a resolver. En primer lugar, elegiremos el grafo en cuestión (1 para el grafo de 6 nodos y 2 para el grafo de 19). Tras elegirlo, se nos abrirá una nueva ventana en la que lo podremos visualizar:

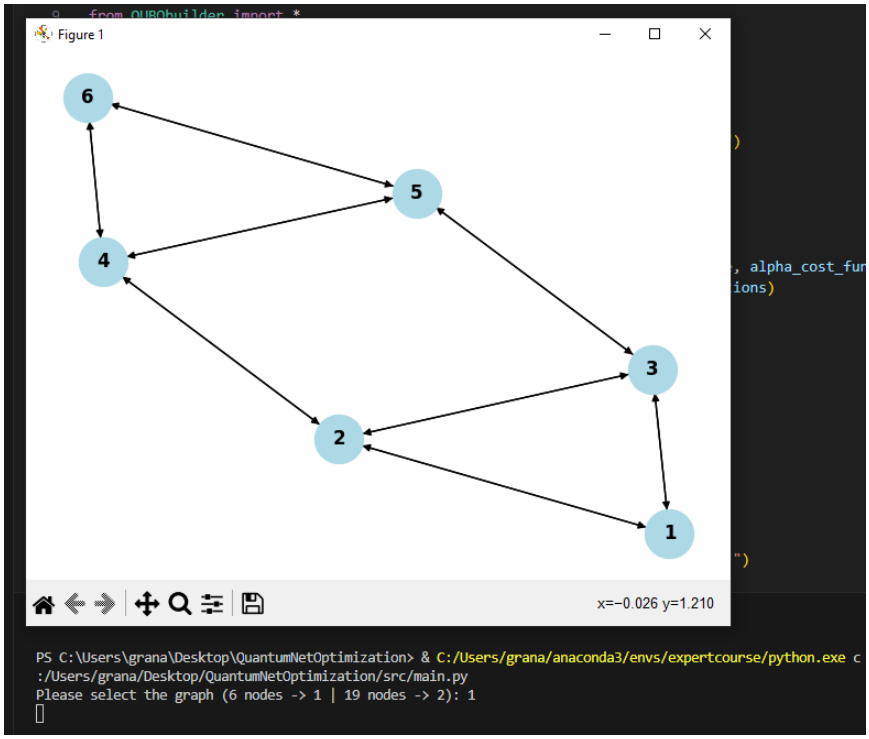

Figura 3: Grafo de 6 nodos.

Para continuar cerramos la ventana. Se pedirá ahora introducir las funciones a solicitar por cada agente, separadas por espacios. Si no queremos ninguna función basta con continuar pulsando enter sin introducir nada. Después se pedirá que introduzcamos el número de agentes o conexiones, siendo el mínimo 1 y el máximo 3. A continuación se nos preguntará por el nodo inicial y el nodo final, comenzando el primer nodo del grafo en 1 y acabando el último nodo en N siendo N el número de nodos del mismo. Por último, se preguntará si se desea ejecutar primero el simulador o si se prefiere pasar directamente al solver híbrido. Como se aclaró en la memoria, no es recomendable ejecutar el simulador en grafos grandes o con muchas

variables y agentes puesto que solo sirve para hacer una primera aproximación al problema.

```
Please select the graph (6 nodes -> 1 | 19 nodes -> 2): 1
Please enter the desired functions, separated by spaces: 3 4
Please enter the number of agents (between 1 and 3): 3
Please enter the start node (starting at 1): 1
Please enter the end node (ending at N): 6
Do you want to execute the simulator first? (not recommended for big graphs, it may be inaccurate due to pr
ocessing limitations)  (1 - yes / 0 - no): 1
```

Figura 4: Terminal durante la ejecución.

Para el caso de la figura 7 se ha escogido el primer grafo, con las funciones 3 y 4. Nodo inicial 1 y final 6, y 3 agentes. Además se ha decidido ejecutar el simulador primero. Tras esto, se obtendrá el resultado del simulador en una nueva ventana:

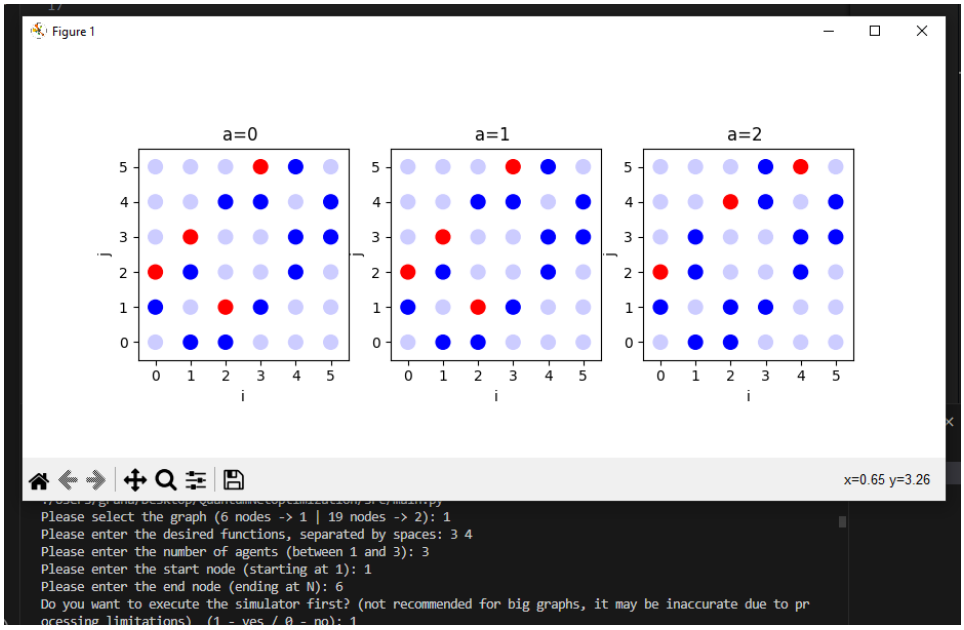

Figura 5: Resultado del simulador.

Para continuar y pasar a ejecutar el solver cerramos la ventana. En caso de no haber ejecutado el simulador este paso no será necesario. Cuando

ejecutemos el solver obtendremos un mensaje en la terminal en el momento que se haya mandado la petición a la plataforma D-Wave Leap.

```
Executing Hybrid Solver, please wait...
```

Figura 6: Mensaje de la terminal.

El resultado del solver será el siguiente:

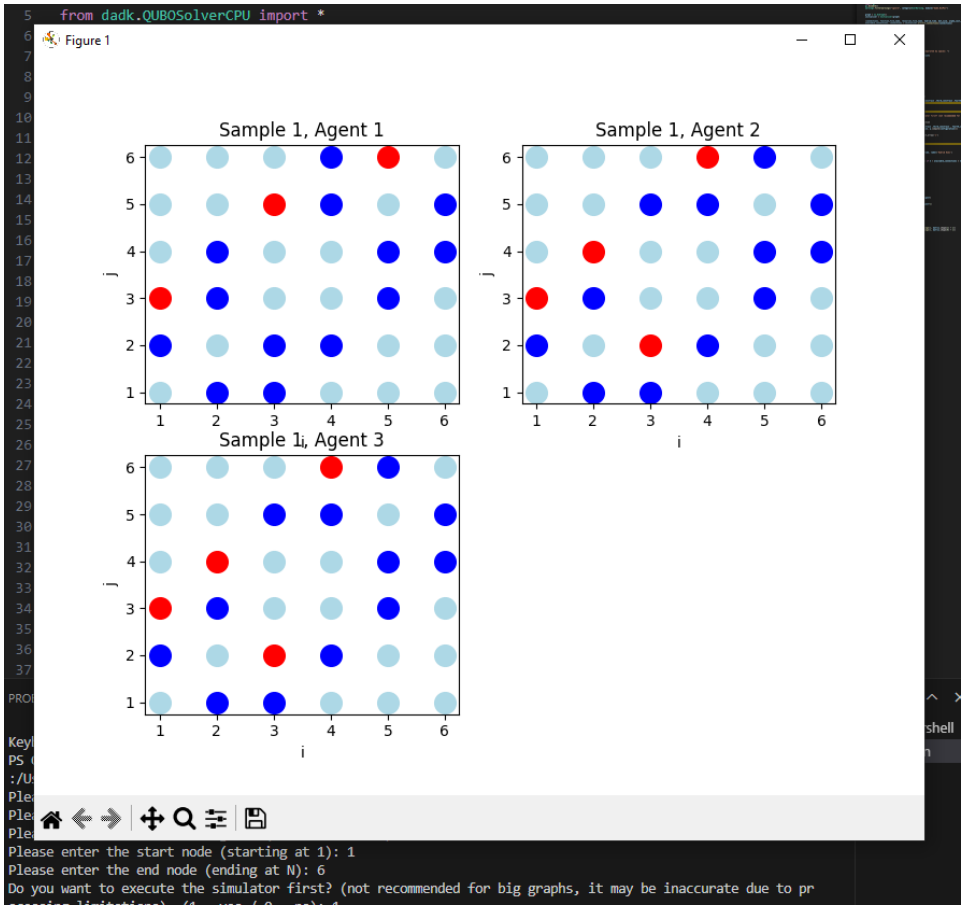

Figura 7: Resultado del solver.

Para poder ejecutar el solver será necesario contar con un token de acceso a la plataforma, para lo cual es necesario tener una cuenta en la misma.

Cada cuenta dispone de 1 minuto de uso de la unidad cuántica, después de agotar el tiempo de uso se deberá renovar el tiempo a través de alguna de las ofertas que disponen, o vincular la cuenta a un repositorio de GitHub para que se renueve este límite de forma mensual.

Para facilitar la corrección de este trabajo se ha definido un token en el archivo de constantes correspondiente que será el que esté en uso. Además comentados se encontrarán otros 2 tokens por si fueran necesarios.

Todo lo relacionado con el desarrollo del código de este trabajo, junto con la memoria y este manual de instalación se encuentran en mi GitHub: MarioGuisado.