

Proyecto de Matematica Aplicada

Mauricio Salim Mahmud Sánchez C312

Abstract

Bot de telegram para la ayuda y resolución de problemas pre-definidos de la asignatura de Modelos de Optimización.

Descripción basica

Si el existe un solucionador de un problema en el bot, el usuario manda la solución obtenida de este problema en un formato establecido (mas abajo), la solución es comparada con la mejor solución, basado en la "distancia" entre ambas soluciones, al usuario se le retorna un *output* de texto con información de "que tan bien estaba su solución" o si se encontró algún error que imposibilitara al solucionador comparar su solución.

Acciones disponibles

1. Comando */start*: Inicia el bot.
2. Comando */help*: Ayuda del bot.
3. Comando */enum*: Enumera todos los solucionadores de problemas que contiene el bot.
4. Comando basado nombre de un solucionador especifico (ej. */got1*): estos comandos se conocen al usar el comando */enum*, envía la información disponible sobre el solucionador en cuestion.
5. Creador de JSONs basado en el formato de un problema, para ayudar al usuario con el formato requerido por los solucionadores.
6. Recibir un archivo de texto con el formato JSON especificado y enviar una respuesta acorde a este.
7. Recibir un mensaje de Telegram con el formato JSON especificado y enviar una respuesta acorde a este.

Uso del bot

1. El estudiante accede al bot, el cual se iniciará con el comando `/start`.
2. Luego se puede acceder a los problemas disponibles usando el comando `/enum`.
3. Se mostrará una lista con los problemas y un comando asociado a cada uno de ellos, al precionar este comando se obtendrá información acerca de este problema.
4. Al abrir información de un problema, además de la explicación acerca de este, se muestran dos botones.
5. El primero muestra como es el formato para mandar una solución al problema.
6. El segundo muestra como es el formato para mandar una solución al problema, con parametros definidos por el usuario.
7. Las soluciones deben tener formato `'json'`, esta solución se puede enviar en un archivo o en un mensaje de telegram.
8. Una vez enviado la solución del problema se retornará un conjunto de mensajes los cuales tienen como objetivo hacer conocer al estudiante que tanto se acerca su solución a la optima de manera personalizada, y también si tuvo algún error al enviarla.

Problemas

Los problemas que contiene el bot los da un proveedor, el proveedor por defecto es uno local ya definido que contiene todos los problemas disponibles en el directorio `'/src/solvers/'`.

La clase abstracta `'BaseProvider'` da la posibilidad de definir otros proveedores de problemas que hereden de esta y establecerlos en el archivo `'src/config.py'`.

Con esto damos la posibilidad de que se establezcan funetes de problemas no necesariamente locales.

Añadir problemas por defecto

Añadir problemas al bot depende del proveedor de problemas, ya que este puede incluso ser un servicio online, pero para el proveedor por defecto se

debe crear una clase que herede de la clase abstracta ‘BaseSolver‘ y rellenar con la información asociada al problema y los metodos de resolución computarizada de este.

El solver debe tener los paramentros de clase *_name*, *_title*, *_text*, *_default_parameters* y *_variables_descriptions* asignados, a demás de tener los metodos abstractos *_solve_model* y *_compare_solution* sobre escritos. El metodo *_solve_model* es el encargado de resolver el problema de manera optima, esta solución es dada por el creador del problema. El metodo *_compare_solution* es el encargado de comparar la solución optima ya previamente calculada en *_solve_model* con la dada por el usuario final (los estudiantes).

Tecnologías usadas

1. Python.
2. Libreria Python-Telegram-Bot para la conexión con Telegram.
3. Libreria Gekko para la resolución de modelos de optimización.