Características y restricciones del Problema de Ruteo de Autobuses Escolares (SBRP)

* Una sola escuela destino de todos los estudiantes.
* Todos los estudiantes son homogéneos y ninguno requerirá tratamiento especial.
* Flota homogénea donde todos los autobuses tienen igual capacidad de trasportación y dicha capacidad es fija.
* Cada parada debe ser visitada como máximo solo una vez.
* Cada estudiante debe poder alcanzar la parada a la que fue asignado.
* En cada ruta, no se debe exceder la capacidad del autobús.
* Cada parada a la que se asigna algún estudiante, debe ser visitada por las rutas de autobuses una y solo una vez.

Definiciones de las clases

* Student: Representa un estudiante. Tiene un id, un name y un par de coordenadas (coord\_x, coord\_y). También tiene una variable assigned\_stop que contendrá la parada asignada a cada estudiante.
* Stop: Representa una parada de autobús. Tiene un id, un name y un par de coordenadas (coord\_x, coord\_y). También tiene una variable num\_assigned\_students que es un contador que mide cuántos estudiantes han sido asignados a cada parada.
* Bus: Representa un autobús individual en la flota. Tiene un id.
* Route: Representa una ruta de autobús. Contiene un bus y una lista de stops.
* School: Representa la escuela a la que los estudiantes están yendo. Tiene un id, un name, y un par de coordenadas (coord\_x, coord\_y).
* SBRP: Representa una instancia del Problema de Ruteo de Autobuses Escolares. Tiene una school, una lista de stops, una lista de students, una lista de routes, una max\_distance que los estudiantes pueden recorrer, una bus\_capacity que es la capacidad de cada autobús, y una student\_stop\_cost\_matrix que es una matriz de costos entre cada estudiante y cada parada.

Métodos para asignar estudiantes a paradas

Se definieron dos métodos para asignar estudiantes a paradas: studentToStop() y studentToRandomStop(). Estos métodos asignan a cada estudiante a una parada que esté dentro de la distancia máxima que el estudiante puede recorrer, ya sea la parada más cercana o una parada aleatoria.

Método para leer datos de una instancia desde un archivo

Se definió un método from\_file() en la clase SBRP que lee los datos de una instancia del problema desde un archivo y crea una nueva instancia de SBRP a partir de estos datos.

Clase Utils

Se creó una clase Utils con dos métodos estáticos: calculate\_distance() y calculate\_cost\_matrix(). calculate\_distance() calcula la distancia euclidiana entre dos conjuntos de coordenadas, y calculate\_cost\_matrix() utiliza este método para calcular una matriz de costos entre dos listas de entidades (que podrían ser estudiantes, paradas, etc.).