Homework #6
Search Tool (Version 2)

AI Programming Due: Nov. 9, 2021 15:00

Overview

Your task in this assignment is to revise the programs you completed in the previous assignment to new versions in an object-oriented programming style. For this, you need to define classes and convert many of the functions to methods of the classes. Followed by the revision, you implement another search algorithm, namely, gradient descent, and add it to your suit of search algorithms.

Defining Classes

It seems natural to define two classes: 'Numeric' and 'Tsp'. In doing so, you can define variables for each class to store the specifics of the problem to be solved. You also need to define methods for setting these variables to their due values. Notice that functions in the previous 'numeric' and 'tsp' modules can become methods of 'Numeric' and 'Tsp' classes, respectively. Perhaps more functions in the main programs may also be migrated to appropriate classes.

When you report the result of search, you always report three things: the solution found, its objective value, and the total number of evaluations taken. Therefore, you can define a new 'Problem' class and make it a parent of both 'Numeric' and 'Tsp', and let it have the variables to store those results. Although the things to be reported are the same—the solution, its objective value, and the total number of evaluations,—the ways they are displayed may be different depending on whether the problem solved is of the type 'Numeric' or 'Tsp'. You should carefully think about how the codes for reporting the result of search can be distributed to different classes within the class hierarchy taking advantage of the concept of inheritance.

It is recommended that you store the class hierarchy of 'Problem' (i.e., 'Problem' with its subclasses 'Numeric' and 'Tsp') in a separate file named 'problem.py'.

Adding Gradient Descent

The gradient descent algorithm has parameters such as the size of increment for calculating derivatives and the update rate used in the gradient update rule. These values can be stored in some class variables, which you need to determine. Since your programs are now object-oriented, you may need to add more methods to relevant classes to support the gradient descent search. Note that gradient descent is not applicable to TSPs.

Calculation of Gradient

Recall the mathematical definition of the derivative as

$$\frac{df(x)}{dx} = \lim_{dx \to 0} \frac{f(x+dx) - f(x)}{dx}$$

Thus, at any specific value of x, we can numerically approximate the derivative as follows:

$$\frac{f(x+\varepsilon)-f(x)}{\varepsilon}$$

where ε is set to a small constant, say around 10^{-4}.

Given a d-dimensional function $f(\boldsymbol{x})$ where $\boldsymbol{x} = (x_1, x_2, \ldots, x_d)^T$, its gradient $\nabla f(\boldsymbol{x})$ is the following vector:

$$\left(\frac{\partial f(\boldsymbol{x})}{\partial x_1}, \frac{\partial f(\boldsymbol{x})}{\partial x_2}, \cdots, \frac{\partial f(\boldsymbol{x})}{\partial x_d}\right)^T$$

The i-th partial derivative in the above vector can be approximately calculated as

$$\frac{\partial f(\boldsymbol{x})}{\partial x_i} = \frac{f(\boldsymbol{x}')-f(\boldsymbol{x})}{\delta}$$

where $\boldsymbol{x}' = (x_1, \ldots, x_{i-1}, x_i + \delta, x_{i+1}, \ldots, x_d)^T$.