MSCI 534 Lent 2023 Coursework

1 Coursework Description

In this coursework, you are to write a program that attempts to solve the set covering problem using heuristic techniques. The set covering problem considers n subsets of a set of m elements. Its objective is to select some of those subsets, so that the collection of subsets selected contains all m elements of the set and its total cost is minimum. The problem can be mathematically stated as $\min c^T x$ subject to $Ax \geq 1$ where A is a binary matrix of size $m \times n$ and $a_{ji} = 1$ if and only if item j is in subset i. x is therefore a binary vector indicating which subsets have been selected. Instances of data files are available at `http://people.brunel.ac.uk/~mastjjb/jeb/orlib/scpinfo.html` as plain text files (.txt). The format of the files is also described on that website.

Specifically, your program will read in a problem instance from a file and will attempt to produce a solution using constructive and improvement heuristics. Note that the optimal solutions to such intractable problems are not currently known for all instances and so algorithms seen to quickly achieve low cost solutions are viewed more favourably. You are required to submit to Moodle a Python .py file. The name of this file should be your library number (e.g. 12345678.py, where 12345678 is an example library number). You should also submit a document file named 12345678.docx, where 12345678 is an example library number, containing:

1. a short summary of your work listing the tasks that you have completed and outlining all the features you have incorporated, e.g., how your constructive heuristic works, the definition of your neighbourhood and operators as well as how your improvement methods work and any additional tricks you have incorporated;

2. a copy of your Python code. Please do not submit compressed files such as zip and rar files; and do not import any packages apart from pandas, matplotlib, numpy, math, random, tkinter, copy, time, and seaborn;

3. a report on the results of your tests on the instances provided on Moodle as described in Task 5.

Problem description

Task 1: (2 marks) Your first task is to write a Python program that asks the user for the input file name and then reads-in the problem instance
file.

Task 2: (5 marks) Construct an initial solution to this problem by a constructive heuristic.

Task 3: (5 marks) Implement a method that improves the initially generated solution using local search. To do this, you will need to implement an appropriate neighbourhood and an operator to modify your solution. Your program should be run for a fixed number of iterations. Ask the user to input the number of iterations. At the end of the run, your best obtained solution should be saved in a .csv file. The
file spec
ifies the solution (subsets selected), and the last row displays the cost of the solution.

Task 4: (12 marks) Try to make your solver more efficient and powerful, by implementing one of the improvement heuristics simulated annealing or tabu search. This is of course in addition to the local search method implemented in the previous task.

Task 5: (6 marks) I will make a suite of test instances available on Moodle about one week before the submission deadline. Perform your heuristic on each of these test instances, possibly repeatedly, and in your docx file report the best result you have found and what parameters (such as temperatures in the simulated annealing heuristic) you have used to obtain this.

It is a good idea to use a small example to test your code while you develop it. For example you can use the instance available on Moodle. This way you have an idea about whether your heuristics are working.

2 Coursework Submission

The coursework deadline is 10:00AM, Tuesday 9th May, 2023. In accordance with University regulations, work submitted up to three days after the deadline will be classed as late submission (10 marks will be deducted), and after three days as a non-submission. However, if an extension is given then the rule applies from the date of the extension.

3 Plagiarism

According to university rules, instances of plagiarism will be treated very seriously. Penalties are in line with the institutional framework of the University. We will analyse the coursework using plagiarism detection software. In the days following submission, and at a prearranged time, selected students will be asked to spend a few minutes in Teams where you will demonstrate your program to me. I will use this opportunity to ask you to provide information on your program and explain parts of your code.