COMP 5413

Final Exam

Submission Due: 16th June 2020, 11:30 PM

Total Marks: 50

Instructions:

a) **There are total 4 optimization problems and you have to answer every question.**

b) **Please note that for all problems, you need to submit the algorithms (as specified in each problem). PLEASE DO NOT SUBMIT ANY PROGRAMMING CODE. You will not get any points for the programming. Your answer will be evaluated by your algorithmic solutions only.**

c) **Please submit a single pdf file including all answers in D2L submission link.**

d) **If you have any trouble in submitting through D2L please email me the pdf immediately at schoudh1@lakeheadu.ca**

e) **Please note that, NO LATE SUBMISSION WILL BE ACCEPTED.**

1. [25 marks] We have computed **n** data files that we want to store, and we have available **W** bytes of storage. A file **i** has size $\mathbf{w_i}$ bytes and takes $\mathbf{m_i}$ minutes to recompute. We want to avoid as much recomputing as possible, so we want to find a subset of files to store such that

   a) The files have combined size at most **W**.
   b) The total computing time of the stored files is **as large as possible**.

   We can not store parts of files; it is the whole file or nothing. Design an **algorithm** to select such subset of files? Show **an example** that explains your solution.

2. [25 marks] Consider the following scenario:

There are **m** jobs and **n** servers where **m>>n (m is much larger than n)**. Each job needs to be placed in one of the servers to be processed. Each server **j** has a capacity **C_j** **(**a server can process maximum **C_j** jobs**)**. If a job **i** placed in a server **j** then it takes **t_{ij}** time to process the job. Our goal is to assign **m** jobs in **n** servers in such a way so that we can minimize the maximum processing time of a job while maintaining the following constraints:

a) No server can process jobs more than its capacity.

b) A job can be assigned to only one server.

c) All jobs need to be processed.

Write down a local search-based algorithm to solve this problem. Analyze the time complexity of the algorithm. Please show an example that explains your solution.

3. [35 marks] Consider the following scenario:

You have one bipartite graph where the size of one partition is higher than the other partition. For each edge of this bipartite graph, there are two weights associated (there are no relationships between these two weights). We can call these two weights as $\mathbf{C_{ij}}$ and $\mathbf{R_{ij}}$ where (i,j) is an edge. It means that if i and j are matched at the end, we need to pay a cost $\mathbf{C_{ij}}$ but we will get some reward as $\mathbf{R_{ij}}$.

The goal is to match these vertices from two partitions in such a way so that we pay the minimum cost while the total sum of rewards that we will get from these matching is more than a threshold **Z**. Please note that, one node from one partition can be **maximum** matched with one node of another partition.

Now, first write down a MILP formulation for this problem and then provide any other algorithmic solution (for example, greedy, local search etc.) to solve this problem.

4. [15 marks] Consider a graph $G = (V, E)$. Write an algorithm (you can use any optimization method) to find a smallest subset $\boldsymbol{D}$ of $\boldsymbol{V}$ such that every vertex not in $\boldsymbol{D}$ is adjacent to at least one member of $\boldsymbol{D}$.