# CSE6140 Assignment 4

due Nov 19, 2017 at 6pm on T-Square

(First submission due Nov 17, updates can be done afterwards until deadline)

**Please read and follow all instructions carefully.**

## Submission Instructions

- You should submit 2 separate files on TSquare:
  - A single typed PDF for the cover letter titled `<GTusername>_HW4_cover_letter.pdf` (e.g. `agupta375_HW4_cover_letter.pdf`), naming any students you collaborated with and any **permitted** resources used.
  - A single typed PDF with your solutions to all problems, titled `<GTusername>_HW4.pdf` (e.g. `agupta375_HW4.pdf`)
- Do not submit .pngs, .pages files, .docx files, .jpgs, .rtfs or any other formats in nested zipped files. Your solutions should be typed, and should be in PDF files.
- Do not submit .rars, .tars, .tar.bz2s, .tar.gzs, .7zs, or any type of archive file.
- Failure to comply with ALL of these instructions will cause you to lose up to all of the points on this assignment.
- Remember the academic honor code: you are not allowed to copy from other students, you may not use material from prior classes, and Googling questions on the internet is not allowed. Any evidence of cheating will be reported and the proper procedures for academic misconduct will be followed.

## 1 Branch and Bound & Local Search

1. Devise a branch-and-bound algorithm for the Minimum SET COVER problem. This entails deciding:

   (a) What is a subproblem?

   (b) How do you choose a subproblem to expand?

   (c) How do you expand a subproblem?

   (d) What is an appropriate lowerbound?

   Do you think that your choices above will work well on typical instances of the problem? Why?

2. Outline a simple greedy heuristic for the SET COVER problem, and explain why it finds a valid solution and its running time.

3. Imagine you wanted to use a local search method to solve Minimum SET COVER such as Simulated Annealing or Iterated Local Search. Imagine a candidate solution is a subset of the sets that might or might not cover all elements in the Universe set U.
   - What could be a possible scoring function for such candidate solutions?
   - What would be a Neighborhood (or Moves) you would consider using for your local search to move from one candidate solution to other 'nearby' solutions? How many potential neighbors can a candidate solution have under your Neighborhood (using Big-Oh)?
   - Why would you consider adding Tabu Memory and what would be remembered in your Tabu Memory?

## 2 Approximation using Greedy Heuristics

Suppose you are in charge of unloading containers from ships arriving at a port onto trucks. A ship arrives with $n$ containers of weight $w_1, w_2, \ldots, w_n$ tons. Standing on the dock is a set of trucks, each of which can hold 1 ton of weight. (You can assume that none of the containers weighs more than 1 ton, since in this case you would not be able to transport it.) You can stack multiple containers in each truck subject to the weight restriction of 1 ton. Hoping to minimize your truck rental expenses, you aim to minimize the number of trucks needed in order to carry all the containers.

You start with the following greedy strategy. Start with an empty truck and begin loading containers $1, 2, 3, \ldots$ onto it until you get to a container that would overflow the weight limit. Now, declare this truck "loaded" and send it off; then continue the process with a fresh truck.

After trying (and failing) to prove that this greedy algorithm is optimal, you decide to attempt to come up with a counter-example to see if you can discover that your strategy is **not** optimal.

(a) Provide an example of a set of weights and a value of $K$ for which your algorithm does *not* use the minimum possible number of trucks. State the number of trucks used by the algorithm for this example, and the true minimum number of trucks needed.

You realize sadly that your problem is in fact NP-complete. However, this renews your faith in your greedy strategy. You try to come up with an approximation ratio.

(b) Let $W = \sum_{i=1}^{n} w_i$ be the total weight of all $n$ containers. Let $t^*$ be the minimum number of trucks you need to rent to transport the $n$ containers. Provide a lower bound for $t^*$, and argue in support of your bound.

(c) Suppose your greedy solution uses an even number of trucks, $t = 2z$ where $z$ is a positive integer. You consider these $2z$ trucks in consecutive pairs. What is a lower bound for the weight of containers in each consecutive pair? Argue in support of your bound.

(d) Using (b) and (c), prove an approximation ratio of 2 for your algorithm.

(e) Prove the same approximation ratio for the case when your greedy approach uses an odd number of trucks, $t = 2z + 1$.

You realize that you might be sending away trucks that still have room for containers that will be unloaded later on. Hoping to improve your strategy, you decide not to dispatch trucks as they are filled, but to keep them around until all $n$ containers have been loaded. Then, you employ the following greedy strategy. Starting with the first truck, see if the container will fit without overloading it. If so, place the container into the first truck; otherwise, consider each subsequent truck in turn and place the container in the first truck that can accommodate it.

(f) Argue that this approach leaves at most one truck less than half full.

(g) Show that the number of trucks you would use with the second strategy is never more than $\lceil 2W \rceil$.

(h) Prove an approximation ratio of 2 for this strategy.

## 3 Modeling with ILP

Formulate the following problems as integer linear programs. In each case, state how many constraints and variables you used.

(a) The independent set problem: Given an undirected graph $G = (V; E)$, find a maximum independent set, that is, a maximum subset of vertices in which no two vertices are adjacent.

(b) The facility location problem: We are given $n$ potential facility locations and a list of $m$ clients who need to be serviced from these locations. There is a fixed cost $f_j$ of opening a facility at location $j$, while there is a cost $c_{ij}$ of serving client $i$ from facility $j$. The goal is to select a set of facility locations and assign each client to one facility, while minimizing the total cost.

(c) The Sudoku problem: Given is an $n^2 \times n^2$ matrix (for some positive integer $n$), which contains some matrix entries pre-filled with integral values between 1 and $n^2$. The task is to fill the remaining entries with integers from $\{1, 2, ..., n^2\}$, such that each row, each column, and each of the $n^2$ major $n \times n$ submatrices contains each integer 1 through $n^2$ exactly once.