St. Francis Xavier University
Department of Computer Science

CSCI 550: Approximation Algorithms
Assignment 2
Due November 4, 2021 at 11:59pm (Atlantic time)

Assignment Regulations.

- This assignment must be completed individually. It is acceptable to discuss the assignment questions with other students, but you must write your answers individually and without assistance.
- Please include your full name and email address on your submission.
- You may either handwrite or typeset your submission. If your submission is handwritten, please ensure that the handwriting is neat and legible, and that the scanned paper is clear.

[10 marks] 1. Consider the following greedy algorithm for the knapsack problem:

- Sort all items in order of non-increasing ratio of value to size: $v_1/s_1 \geq v_2/s_2 \geq \cdots \geq v_n/s_n$.
- Let i^* denote the index of an item of maximum value; that is, $v_{i^*} = \max_{i \in I} v_i$.
- Add items to the knapsack in index order until the next item does not fit; that is, find k such that $\sum_{i=1}^{k} s_i \leq B$ but $\sum_{i=1}^{k+1} s_i > B$.
- Return either $\{1, \dots, k\}$ or $\{i^*\}$, whichever has greater value.

Prove that this greedy algorithm is a $\frac{1}{2}$-approximation algorithm for the knapsack problem.

Hint. What is an upper bound on the optimal value?

[10 marks] 2. Recall the problem of weighted sequential scheduling on a single machine, and consider a variant of this problem where release dates are not used; that is, given a set of jobs, we must schedule each job on a single machine nonpreemptively in a way that minimizes the weighted sum of completion times $\sum_{j=1}^{n} w_j C_j$.

Suppose that jobs are sorted in order of non-increasing ratio of weight to processing time: $w_1/p_1 \geq w_2/p_2 \geq \cdots \geq w_n p_n$. Prove that our optimal strategy is to schedule job 1 first, job 2 second, and so on.

Hint. Consider what happens if two adjacent jobs are placed out of order and we swap the position of these two jobs in the ordering.

[10 marks] 3. Consider the following variation of the weighted maximum satisfiability problem: the formulation of the problem is the same, but we additionally assume that each clause contains only positive variables and there is a weight $v_i \geq 0$ associated to each Boolean variable x_i. The objective of this problem is to find an assignment of values to each Boolean variable that maximizes the total weight of the satisfied clauses *and* the total weight of Boolean variables set to false.

(a) Give an integer programming formulation for this problem. You may assume that you are given variables y_i that you can use to indicate whether the variable x_i in the Boolean formula is set to true.

(b) What modifications would you need to make to your integer program from part (a) to relax it to a linear program?

[10 marks] 4. Consider the following generalization of the weighted maximum cut problem:

<u>WEIGHTED-MAX-K-CUT</u> Given: an undirected graph $G = (V, E)$ and an associated weight $w_{ij} \geq 0$ for each edge $(i, j) \in E$ Determine: a partitioning of V into k subsets $V_1, \ldots, V_k$ that maximizes $w = \sum_{\substack{i \in V_a, j \in V_b \\ \text{for } a \neq b}} w_{ij}$

In the *weighted maximum k-cut problem*, we must partition the vertex set into k subsets instead of two. The objective remains the same: to maximize the weight of the cut edges.

Give a randomized $\left(\frac{k-1}{k}\right)$-approximation algorithm for the weighted maximum k-cut problem.

Hint. We saw a naïve algorithm for the weighted maximum cut problem that had a performance guarantee of $\frac{1}{2}$. How can we adapt that algorithm to work for this more general problem?