Homework 4

Due On: December 7, 2020 2:59PM (CDT)

Problem 1 (4 pts) Given a graph $G(V, E)$ we want to color the vertices with minimum number of colors such that each pair of adjacent vertices have different colors.

(a) (1.5pts) Develop an integer programming formulation for the graph coloring problem.

(b) (1.5pts) Now consider the following vector program:

$$\begin{aligned} & \min \kappa \\ \text{subject to } & v_i \cdot v_j \leq \kappa \quad \forall (i,j) \in E \\ & ||v_i|| = 1 \quad \forall i \in V \\ & v_i \in \mathbb{R}^n \quad \forall i \in V \end{aligned}$$

Explain why this is a relaxation to the coloring problem. (1.5)

(c) (1pt) For the triangle graph (K_3) determine an optimal solution of the above vector program.

Problem 2. (3 pts) Suppose our cow (from the cowpath problem) is at the center $(0,0)$ of a infinite 2D grid. The grass is located at some point (n, m) on this grid. The cow can only move up/ down or left/right. Devise the strategy where the cow finds the patch of grass with the least effort. What is the competitive ratio of your algorithm? What if instead of the patch of grass there is an infinite length fence (extending in both directions), which could either vertical or horizontal, located somewhere in the grid (e.g. $x = n$ or $y = -n$). Devise an algorithm for this case and analyze its competitive ratio.

Problem 3. (3 pts) Suppose you're consulting for a biotech company that runs experiments on two expensive high-throughput assay machines, each identical, which we'll label M_1 and M_2. Each day they have a number of jobs that they need to do, and each job has to be assigned to one of the two machines. The problem they need help on is how to assign the jobs to machines to keep the loads balanced each day. The problem is stated as follows. There are n jobs, and each job j has a required processing time t_j. They need to partition the jobs into two groups A and B, where set A is assigned to M_1 and set B to M_2. The time needed to process all of the jobs on the two machines is $T_1 = \sum_{j \in A} t_j$ and $T_2 = \sum_{j \in B} t_j$. The problem is to have the two machines work roughly for the same amounts of time, that is, to minimize $|T_1 - T_2|$.

Consider the following algorithm. Start by assigning jobs to the two machines arbitrarily (say jobs 1.....$n/2$ to M_1, the rest to M_2). The local moves are to move a single job from one machine to the other, and we only move jobs if the move decreases the absolute difference in the processing times. You are hired to answer some basic questions about the performance of this algorithm.

(a) How good is the solution obtained? Assume that there is no single job that dominates all the processing time, that is, that $t_j < \frac{1}{2}\sum_{i=1}^{n} t_i$; for all jobs j. Prove that for every locally optimal solution, the times the two machines operate are roughly balanced: $\frac{1}{2}T_1 \leq T_2 \leq 2T_1$.

(b) What can you say about the running time of this algorithm? Suggest a modification that improves the running time.