Problem Set no. 3

Given: May 11, 2018
Due: May 25, 2018

Exercise 2.1 Consider the example given in class that shows that the local search for max cut in a weighted graph takes exponential time. The construction, also shown in Figure 1, used vertices of two kinds (Please review the definitions on slides 24-31). The vertices $v_1, \ldots v_n$ are of the second kind and all the other vertices are of the first kind.
a) Assign **integer** weights to the edges of the graph so that all the vertices are of the kind they need to be (note that the formula in reference 4 in the bibliography list is incorrect, so do not use it).
b) Define recursively a sequence of improvement moves that make v_n flip 2^{n+1} times and prove by induction that v_n indeed flips 2^{n+1} times (this is essentially what we did in class, write it down carefully and formally to make sure you understand it.)

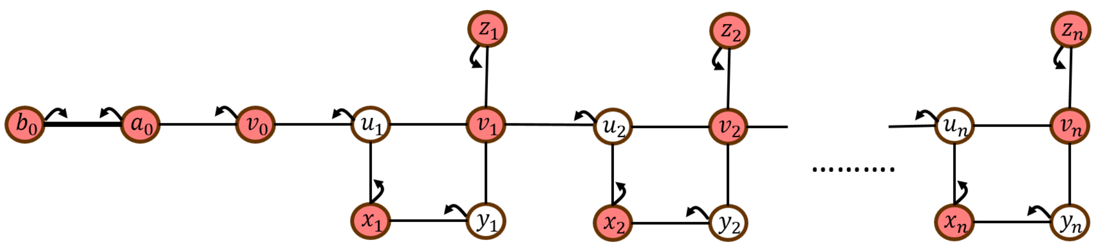

Figure 1: Weighted graph used in the lower bound example for max cut

Exercise 2.2 Consider the algorithm of Lin and Kernighan for minimum bisection. Suppose that G is unweighted and that when finding the next pair to match we do not necessarily insists on the one that decreases the bisection the most (or increases the least), but we allow for an additive error of -2. That is the next pair which we match decreases the bisection by at least $A - 2$ where A is the decrease of the best pair. Show how to implement a step of this version of the algorithm in $O(m)$ time where m is the number of edges in the graph. (By a step we mean computing the matching and finding an improving neighbor or concluding the none exists. Note that A may be negative.)

Exercise 2.3 Consider the α-expansion local search algorithm presented in class. Let OPT be an optimal solution and assume that $\sum_{(v,w)\in E} p(v,w) \le \epsilon OPT$ for some $\epsilon \le 1$ (We use OPT here to denote both the optimal solution and its value). Prove that the value of local minimum returned by the α-expansion procedure is at most $(1+\epsilon)OPT$. Please prove this rigorously and repeat details mentioned in class that are required for the proof.

Exercise 2.4 We have a set of m jobs and 2 machines. Job j has integer length w_j, and $\sum w_j = W$. We would like to assign each job j to a machine $m(j) \in \{1,2\}$ such that the maximum load on a machine is minimized. That is, we want to find an assignment that minimizes $\max\{\sum_{j|m(j)=1} w_j, \sum_{j|m(j)=2} w_j\}$. We define the neighbors of an assignment m to be any assignment m' that we can obtain from m by moving one job from one machine to the other or swapping

two jobs, one from each machine. We run a local search procedure using this neighborhood relation starting from an arbitrary assignment.
a) Give the best upper bound that you can on the number of times this algorithm either moves or swaps jobs.
b) Prove that the maximum load of a machine in the local minimum is at most $\frac{4}{3}OPT$ where OPT is the maximum load of a machine in the optimal assignment.
c) Is the bound stated in part (b) tight ?

Exercise 2.5 Given a set P of n points. a) Prove that for any set $F \subseteq P$, $|F| = k$, there exists a pair $x \in F$, $y \in P$, such that

$$cost(F) - cost(F \setminus \{x\} \cup \{y\}) \geq \frac{cost(F) - 5OPT}{k} ,$$

where $cost(A)$, $|A| = k$, is the sum of the distances of the points of P from the centers in A, and OPT is the cost of the best (according to the k-medians objective) k centers in P.
b) Suppose that the distance between any pair of points of P is at least 1 and at most Δ. We run the local search algorithm (for k-median) presented in class for $O(k \log(n\Delta))$ iterations, such that we choose the most improving swap in each iteration. Prove that the cost of the local solution we reach is at most $6OPT$.
(Hint: Prove that if $cost(L) \geq 6OPT$ in some iteration, where L is the local solution, then after the next swap we have that $cost(L') \leq (1 - \frac{1}{6k})cost(L)$, where L' is the local solution obtained from L by the swap.)