Problem Set no. 2

Given: May 1, 2017
Due: May 17, 2017

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 $v_1, \dots v_n$ 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.)

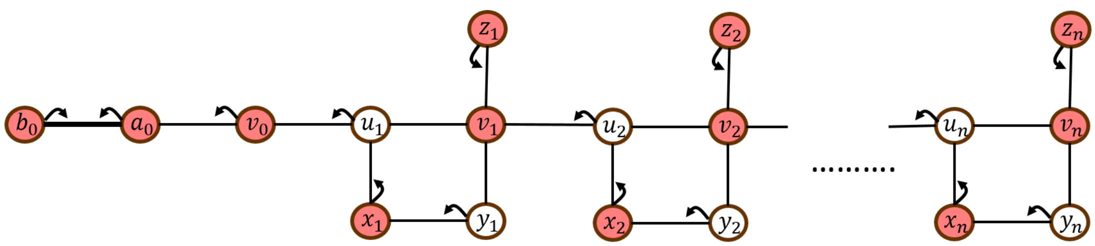

Figure 1: Weighted graph used in the lower bound example for max cut

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 Consider the k-medians problem. Let $A = \{1, \ldots, n\}$ be a set of points which are input to the k-medians problem and let d denote the metric. Fix a global optimal solution OPT and some other solution L. For $j \in A$ let $\phi(j)$ the center closest to j in OPT and $\eta(j)$ the center closest to j in L.

a) Prove that $d(j, \eta(\phi(j))) \leq 2d(j, \phi(j)) + d(j, \eta(j))$.

Consider now a local search algorithm where the neighbors of a set S of k-centers is any set S' of k-centers that can be obtained from S by swapping $y \leq t$ centers of S with y points in $A \setminus S$. Assume that L now denotes a local optimum with respect to a swap of a set of size $\leq t$.

We also need the following definitions. For $r \in L$, let $\eta^{-1}(r) = \{j \mid \eta(j) = r\}$ and for $f \in OPT$, let $\phi^{-1}(f) = \{j \mid \phi(j) = f\}$. For $R \in L$, let $\eta^{-1}(R) = \cup_{r \in R}(\eta^{-1}(r))$. Similarly, for $F \in OPT$, let $\phi^{-1}(F) = \cup_{f \in F}(\eta^{-1}(f))$.

For $r \in L$ we denote $\text{opt}(r) = \eta^{-1}(r) \cap OPT$. This is the subset of OPT consisting of all the centers in OPT whose closest center in L is r. We also define $\deg(r) = |\text{opt}(r)|$.
In order to analyze this algorithm we *pair* subsets of L with subsets of OPT as follows. Pick some $r \in L$ such that $\deg(r) > 0$. Pick $\deg(r) - 1$ elements $r_1 \ldots r_{\deg(r)-1}$ of L such that $\deg(r_i) = 0$. Pair $\text{opt}(r)$ with $\{r, r_1, \ldots, r_{\deg(r)-1}\}$. Remove $\text{opt}(r)$ from OPT and $\{r, r_1, \ldots, r_{\deg(r)-1}\}$ from L and repeat.

b) Prove that every time this algorithm needs to pick $\deg(r) - 1$ elements of L with degree 0 then there are such elements in L. Let (F_i, R_i), $i = 1, \ldots, p$, be the pairs that the algorithm generates, where $F_i \subset OPT$ and $R_i \subset L$. Prove that $F_i \cap F_j = \emptyset$ and $\cup_{i=1}^{p} F_i = OPT$, and similarly that $R_i \cap R_j = \emptyset$ and $\cup_{i=1}^{p} R_i = L$.
c) Prove that if $\phi(j) \notin F_i$ then $\eta(\phi(j)) \notin R_i$.
d) Prove that if $|R_i| = |F_i| \leq t$ then

$$\text{cost}((L \setminus R_i) \cup F_i) - \text{cost}(L) \leq \sum_{j \in \phi^{-1}(F_i)} (d(j, \phi(j)) - d(j, \eta(j))) + \sum_{j \in \eta^{-1}(R_i)} 2d(j, \phi(j)) \ .$$

e) Consider now the case where $|R_i| = |F_i| = s > t$. Let $R'_i \subset R_i$ denote the points in R_i that had degree 0 when we defined R_i. Note that $|R'_i| = s - 1$. Prove that

$$\sum_{(f,r) \in F_i \times R'_i} (\text{cost}(L \setminus \{r\} \cup \{f\}) - \text{cost}(L)) \leq (s-1) \sum_{j \in \phi^{-1}(F_i)} (d(j, \phi(j)) - d(j, \eta(j))) + s \sum_{j \in \eta^{-1}(R'_i)} 2d(j, \phi(j)) \ .$$

f) Conclude that $\text{cost}(L) \leq (3 + 2/t)\text{cost}(OPT)$. (Note that $s/(s-1) \leq 1 + 1/t$ since $s > t$.)