Design and Analysis of Algorithms, Fall 2014
Exercise VI: Solutions

VI-1 (**CLRS 26.3-4**$\star$) A *perfect matching* is a matching in which every vertex is matched. Let $G = (V,E)$ be an undirected bipartite graph with vertex partition $V = L \cup R$, where $|L| = |R|$. For any $X \subseteq V$, define the *neighborhood* of X as $N(X) = \{y \in V : (x,y) \in E \text{ for some } x \in X\}$, that is, the set of vertices adjacent to some member of X. Prove *Hall's theorem*: there exists a perfect matching in G if and only if $|A| \leq |N(A)|$ for every subset $A \subseteq L$.

Let's first prove the direction: if there exists a perfect matching in G, then $|A| \leq |N(A)|$ for every subset $A \subseteq L$.

A perfect matching in G means that every vertex of L is connected to another vertex of $|R|$, exclusively. Besides, a vertex of L can have more incident edges. Therefore, $|A| \leq |N(A)|$.

Now let's look at the other direction of the proof: if $|A| \leq |N(A)|$ for every subset $A \subseteq L$, then there exists a perfect matching in G.

When $|L| = |R| = 1$, the base case, it is easy to see that there is a perfect matching, i.e. the only edge in G.

Suppose that the above claim is valid for $|L| = |R| = 1, 2, ..., n-1$.

When $|L| = |R| = n$, we consider two separate cases.

Case I: $|A| < |N(A)|$ for every subset $A \subset L$. Pick an arbitrary vertex u in L and one of its neighbors v in R, and remove them from the graph. In the remaining graph G', we must still have $|A| \leq |N(A)|$ for all A. Since $|L| = |R| = n-1$ in G', by the induction hypothesis, there is a perfect matching M'. Consequently, adding the edge (u,v) to M' results in a perfect matching in G.

Case II: there exits at least a subset $A \subset L$ such that $|A| = |N(A)|$. By the induction hypothesis, there is a perfect matching M_A between A and $N(A)$. Now, remove A and $N(A)$ from G. In the remaining graph G', we still have $|B| \leq |N(B)|$ for every subset $B \subseteq L \setminus A$. This can be proved by contradiction. Suppose that $|B| > |N(B)|$. Then, $|A \cup B| > |N(A \cup B)|$ in G. Thus, G' has a perfect matching M'. The union of M_A and M' gives a perfect matching in G.

VI-2 (**CLRS 35.1-3**$\star$) Professor Bündchen proposes the following heuristic to solve the vertex-cover problem. Repeatedly select a vertex of highest degree, and remove all of its incident edges. Give an example to show that the professor's heuristic does not have an approximation ratio of 2. (*Hint:* Try a bipartite graph with vertices of uniform degree on the left and vertices of varying degree on the right.)

Consider a bibartite graph with left part L and right part R such that L has 5 vertices of degrees $(5,5,5,5,5)$ and R has 11 vertices of degrees $(5,4,4,3,2,2,1,1,1,1,1)$ (the graph is easy to draw and the figure is omitted here). Clearly there exists a vertex-cover of size 5 (the left vertices). The idea is to show that the proposed algorithm chooses all the vertices on the right part, resulting in the approximation ratio of 11/5 > 2. After choosing the first vertex in R, the degrees on L decrease to $(4,4,4,4,4)$. After choosing the second vertex in R, the degrees on L decrease to $(4,3,3,3,3)$. After choosing the third vertex in R, the degrees on L decrease to $(3,3,2,2,2)$. After choosing the fourth vertex in R, the degrees on L decrease to $(2,2,2,2,1)$. After choosing the fifth vertex in R, the degrees on L decrease to $(2,2,1,1,1)$. After choosing the sixth vertex in R, the degrees on L decrease to $(1,1,1,1,1)$. Now the algorithm still has to choose 5 more vertices.

VI-3 (**CLRS 35.2-2**) Show how in polynomial time we can transform one instance of the traveling-salesman problem into another instance whose cost function satisfies the triangle inequality. The two instances must have the same set of optimal tours. Explain why such a polynomial-time transformation does not contradict Theorem 35.3, assuming that P $\neq$ NP.

An instance I of the traveling-salesman problem consists of n cities and n^2 edges between pairs of cities. We transform I into another instance I' that satisfies the triangle equality in the following way. Let m be the maximal edge cost among all pairs of cities. Add m to the edge cost of every pair of cities, which becomes I'. The time for constructing I' is $O(n^2)$, which is polynomial.

Now we prove that I' satisfies the triangle inequality. Consider any three cities, with edge cost a, b, c between them. We have

$$\begin{aligned} a+m &\leq a+m+b+c && \text{(edge cost is nonnegative)} \\ &\leq m+m+b+c && \text{(m is the maximal edge cost)} \\ &= (b+m)+(c+m)\ . \end{aligned}$$

Still, we have to prove that I and I' have the same set of optimal tours. Suppose that the optimal tour T in I has cost c. Then, the cost of T in I' is $c+mn$. Assume that there is a better tour T' in I', with cost c', $c' < c+mn$. Then, the cost of T' in I is $c'-mn$. We have

$$\begin{aligned} c'-mn &< (c+mn)-mn \quad \text{(by } c' < c+mn\text{)} \\ &= c\ . \end{aligned}$$

This contradicts the assumption that T being the optimal tour in I. For the case that there is a worse tour T' in I', similar proof can be derived. Together, we showed that the optimal tours are preserved in I'.

Finally, we prove that the above transformation does not contradict Theorem 35.3. In general, a constant-factor approximation to the transformed instance does not guarantee any constant-factor approximation to the original (non-metric) instances. It is because the term mn may dominate the optimal cost of the transformed instance, as we will show below. Assume that there is a ρ'-approximation algorithm A' for I', $\rho' \geq 1$. We run A' on I' and A' outputs a tour T' which has cost c', where $c' \leq \rho' c'^*$ and c'^* is the cost of the optimal tour in I'. Since the optimal tours in both I and I' are the same, as just proved, the cost of the optimal tour in I is c^* and $c'^* = c^* + mn$. The cost of a tour output by A' in I is c and $c' = c+mn$. We want to show that it may happen $\frac{c}{c^*} > \rho$, even though $\frac{c+mn}{c^*+mn} \leq \rho'$ for any constant $\rho \geq 1$ and $\rho' \geq 1$. The key idea is to consider an instance where c^* is very small compared to m. For instance, let

$$\begin{aligned} &c^* = n\ , \\ &c = (\rho+1)n \quad \text{(hence, } \tfrac{c}{c^*} > \rho\text{)}, \\ &m = c^* X = nX \quad \text{(where X will be chosen to be large enough)}\ . \end{aligned}$$

Now, $\frac{c+mn}{c^*+mn} = 1 + \frac{\rho}{1+nX}$ and thus less than ρ' as soon as we choose an X large enough.

VI-4 (**CLRS 35.2-5**) Suppose that the vertices for an instance of the traveling-salesman problem are points in the plane and that the cost $c(u,v)$ is the euclidean distance between points u and v. Show that an optimal tour never crosses itself.

We show that eliminating a crossing always results in a tour with smaller cost. In the figure below, we eliminate the crossing by replacing the edge AD with AB and the edge BC with CD. According to the triangle inequality of euclidian distance, we have

$$c(AB) < c(AO) + c(OB) \quad (1)$$

and

$$c(CD) < c(CO) + c(OD) \quad (2)$$

By (1) + (2), we obtain $c(AB) + c(CD) < c(AD) + c(BC)$.

Therefore, the cost of the newly constructed tour is smaller than the cost of the tour with crossing.

VI-5 (**CLRS 35.4-3**) In the MAX-CUT problem, we are given an unweighted undirected graph $G = (V,E)$. We define a cut $(S, V-S)$ as in Chapter 23 and the weight of a cut as the number of edges crossing the cut. The goal is to find a cut of maximum weight. Suppose that for each vertex v, we randomly and independently place v in S with probability $1/2$ and in $V-S$ with probability $1/2$. Show that this algorithm is a randomized 2-approximation algorithm.

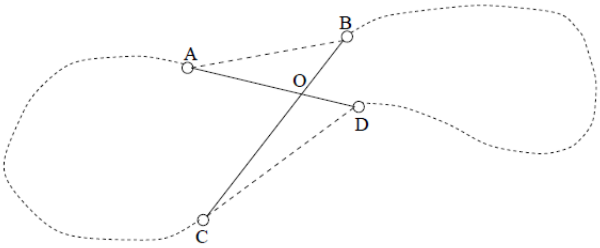

Figure 1: A tour crossing itself.

Suppose that for each vertex v, we randomly and independently place v in S with probability $1/2$ and in $V-S$ with probability $1/2$. For an edge e_i, we define the indicator random variable $Y_i = I\{e_i$ crossing a cut$\}$. For an edge e_i being crossing a cut, its two vertices u, v have to be in S and $V-S$ separately. The probability of such an event is $Pr\{e_i$ crossing a cut$\} = Pr\{u$ in S and v in $V-S\} \cup \{u$ in $V-S$ and v in $S\} = 1/2 \times 1/2 + 1/2 \times 1/2 = 1/2$, and by Lemma 5.1, we have $E[Y_i] = 1/2$. Let Y be the number of edges crossing a cut, so that $Y = Y_1 + Y_2 + \ldots + Y_n$, $n = |E|$. We have

$$\begin{aligned} E[Y] &= E[\textstyle\sum_{i=1}^{n} Y_i] \\ &= \textstyle\sum_{i=1}^{n} E[Y_i] \quad \text{(by linearity of expectation)} \\ &= \textstyle\sum_{i=1}^{n} \frac{1}{2} \\ &= \tfrac{1}{2}n \; . \end{aligned}$$

On the other hand, let c^* be the weight of the max-cut. The upper bound of c^* is the total number of edges, i.e. $c^* \leq n$. Then, we have

$$\begin{aligned} c^* &\leq n \\ &= 2 \times \tfrac{1}{2}n \\ &= 2E[Y] \; . \end{aligned}$$

That is, $\frac{c^*}{E[Y]} \leq 2$ (note that this is a maximization problem).