CS577: Homework 11

Haruki Yamaguchi
hy@cs.wisc.edu

Keith Funkhouser
wfunkhouser@cs.wisc.edu

December 10th, 2015

(a) Consider that problem described is nearly identical to the one described in Homework 10, except that we have removed the bound of 3 for the number of times that a city can be visited. Now, instead of looking for the minimum weight spanning tree among all spanning trees of maximum degree 3, we are looking for a spanning tree of the graph G (without the degree restriction), such that the maximum of the total effort of Abe and of Honest is minimized.

An NP-complete decision problem (`TT-DEC`) that is equivalent to the given problem under polynomial-time reductions is given below:

Input: A complete graph $G = (V, E)$ with non-negative effort functions $A : E \to [0, +\infty)$ and $H : E \to [0, +\infty)$, for the effort expended by Abe and Honest, respectively; also, a parameter k.

Output: 'Yes' if a spanning tree exists in G where $\texttt{max}\{\text{total effort of Abe, total effort of Honest}\}$ is at most k; otherwise, 'no'.

(b) To show that the decision problem formulated above is NP-complete, we will show:

(i) `PARTITION-SUM` $\leq$ `TT-DEC`

(ii) `TT-DEC` $\in$ NP

The first makes use of the structure of `PARTITION-SUM` in order to show that it reduces to `TT-DEC`. The second is straightforward to show using the definition of `TT-DEC` above.

Recall that `PARTITION-SUM` is a decision problem, formulated as follows: for a set of n integers $a_1, a_2, \ldots, a_n \geq 0$, determine whether there exists a partition $I \subseteq \{1, 2, \ldots, n\}$ such that $\sum_{i \in I} a_i = \sum_{i \notin I} a_i$ (yes or no). For the reduction from `PARTITION-SUM` to `TT-DEC`, given the same set of integers, consider the following graph construction:

(a) For each a_i, create two vertices v_i and v'_i. Let $A(v_i, v'_i) = H(v_i, v'_i) = 0$.

(b) Create a vertex u.

(c) For each $i \in \{1, 2, \ldots, n\}$, let $A(u, v_i) = a_i$ and $H(u, v_i) = 0$.

(d) For each $i \in \{1, 2, \ldots, n\}$, let $A(u, v'_i) = 0$ and $H(u, v'_i) = a_i$.

The final construction is illustrated in Figure 1.

Intuitively, a spanning tree of G will include at least one of the edges from u to v_i or v'_i for all i. We can think of selecting the edge to v_i as putting a_i in Abe's bucket, and likewise selecting the edge to v'_i as putting a_i in Honest's bucket. Note also that the edges between v_i and v'_i will always be included in a spanning tree of G that seeks to minimize the maximum of $\sum A$ and $\sum H$ (unless $a_i = 0$), since not including some (v_i, v'_i) implies that both the edges (u, v_i) and (u, v'_i) are in the tree, which is more costly than simply taking only one of those edges, in addition to (v_i, v'_i).

Thus, for each i, either the edge (u, v_i) or the edge (u, v'_i) is included in the spanning tree, but not both (excluding the case where $a_i = 0$, when both could appear, but would not contribute to the total effort of Abe or Honest). This implies that each $a_i > 0$ will contribute to Abe's total effort (henceforth E_A), or Honest's total effort (E_H), but not both. Furthermore, since we are looking for a spanning tree, by

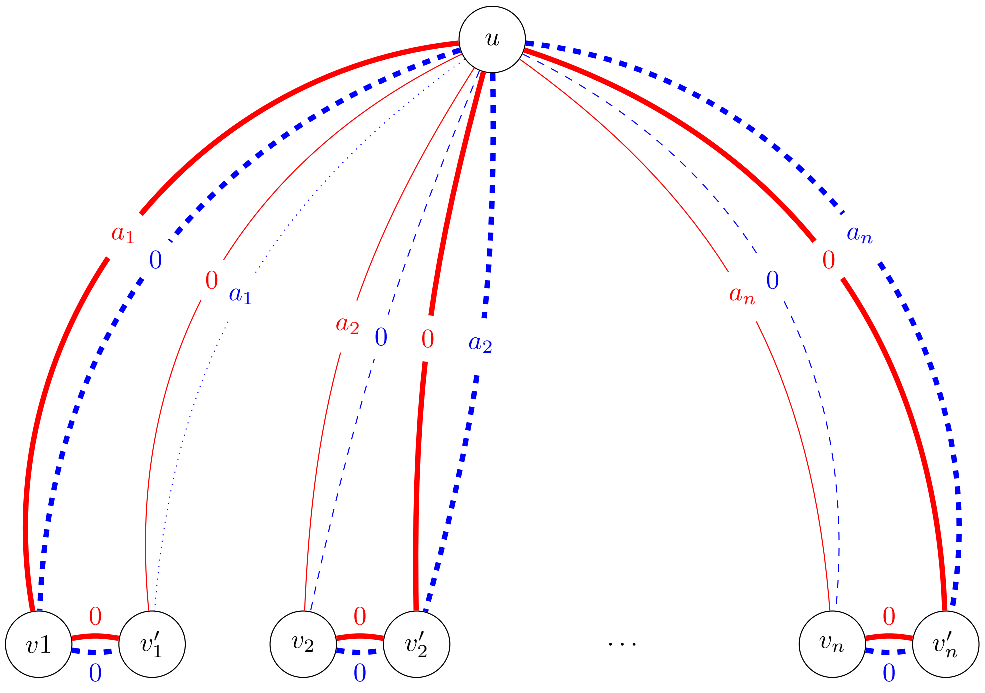

Figure 1: The graph produced by casting an instance of `PARTITION-SUM` into a graph which `TT-DEC` can be run on. Efforts for Abe and Honest are denoted by solid and dashed lines, respectively. One possible spanning tree is shown in bold, which contributes a_1 to Abe's total effort and $a_2 + a_n$ to Honest's. Running `TT-DEC` on G with $k = \frac{\sum_i a_i}{2}$ will return 'yes' iff there exists a valid partitioning of $a_1, a_2, \dots, a_n$.

definition we must reach every vertex, so each a_i will be included. This is exactly a partitioning of the set $a_1, a_2, \dots, a_n$ (i.e. $E_A + E_H = \sum_i a_i$).

Now that we have constructed the graph, it is not entirely obvious what parameter k should be used on input to `TT-DEC`. On inspection, however, the condition in `PARTITION-SUM` can be rephrased:

$$\sum_{i \in I} a_i = \sum_{i \notin I} a_i \Rightarrow \sum_{i \in I} a_i = \sum_{i \notin I} a_i = \frac{\sum_i a_i}{2}$$

(since $\sum_{i \in I} a_i + \sum_{i \notin I} a_i = \sum_i a_i$). A reasonable input k to `TT-DEC` is $\frac{\sum_i a_i}{2}$. To see why, we observe the graph G that we constructed. Consider the case where $\mathtt{max}\{E_A, E_H\}$ is exactly $k = \frac{\sum_i a_i}{2}$. Previously, we showed that $E_A + E_H = \sum_i a_i$, so in this case we have $E_A = E_H = \frac{\sum_i a_i}{2}$ (i.e. `PARTITION-SUM` has a solution for the given instance). In the case where one of $\{E_A, E_H\}$ is greater than $k = \frac{\sum_i a_i}{2}$, then we also know that the other of $\{E_A, E_H\}$ is less than k since $E_A + E_H = \sum_i a_i$. We can make an even stronger statement: we have a solution to `PARTITION-SUM` iff $\mathtt{max}\{E_A, E_H\}$ is *exactly* $k = \frac{\sum_i a_i}{2}$ (proof: suppose there exists a spanning tree with $\mathtt{max}\{E_A, E_H\} < k$. Then $E_A + E_H < 2k = \sum_i a_i$, but we previously showed that $E_A + E_H = \sum_i a_i$).

In short, our reduction is as follows: given an input set of non-negative integers $a_1, a_2, \dots, a_n$, create G as described. Run `TT-DEC` on G with $k = \frac{\sum_i a_i}{2}$. If `TT-DEC` returns 'yes' (i.e. there exists a spanning tree of G with $\mathtt{max}\{E_A, E_H\} = k$, then `PARTITION-SUM` should return 'yes'; otherwise, `PARTITION-SUM` should return 'no'. Since the creation of G can be done in $O(N)$ time in terms of the bit length of the input to `PARTITION-SUM` (since we are simply creating a vertex u and two vertices v_i and v'_i for each a_i), and we have access to the `TT-DEC` oracle at unit cost (we only query it once), The reduction is polynomial in time.

To complete the proof that `TT-DEC` is NP-complete, we now show that `TT-DEC` $\in$ NP. Given inputs to `TT-DEC` (G, A, H, and k) and some potential solution (i.e. a spanning tree of G), it is clear that we can verify whether the spanning tree has $\texttt{max}\{E_A, E_H\} \leq k$ in polynomial time. We traverse the spanning tree and separately sum the efforts for Abe and Honest, take the maximum of the two sums, and check to see if it is greater than k. This can be done in $O(|V|)$ time since the size of the spanning tree will be $|V| - 1$. Since `TT-DEC` $\in$ NP and `TT-DEC` is NP-hard (because `PARTITION-SUM` is NP-complete and `PARTITION-SUM` $\leq$ `TT-DEC`), `TT-DEC` is NP-complete.